



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE ZMĚN V DIGITÁLNÍCH OBRAZECH

DETECTION OF CHANGES IN DIGITAL IMAGES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Dorazil

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Kamil Říha, Ph.D.

BRNO 2017



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Jan Dorazil

ID: 154696

Ročník: 2

Akademický rok: 2016/17

NÁZEV TÉMATU:

Detekce změn v digitálních obrazech

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte principy metod pro detekci změn v digitálních obrazech s cílem detekování změn u objektu zachyceného ve dvou snímcích v různých časech. Prostudujte možnosti současných metod a jejich použitelnost při řešení tohoto problému. Uvažujte především aplikovatelnost v průmyslovém odvětví. Implementujte vybraný algoritmus a ověřte jeho funkčnost na menší databázi dvojic záběrů s definovanými změnami.

Doporučený nástroj pro implementaci: knihovny OpenCV a MS Visual C++.

DOPORUČENÁ LITERATURA:

[1] GONZALEZ, R. C.; WOODS R. E.: Digital Image Processing, Prentice Hall, New Jersey, 2002.

[2] BRADSKI, G.; KAEHLER A.: Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, Inc. USA 2008, ISBN: 978-0-596-51613-0.

Termín zadání: 1.2.2017

Termín odevzdání: 24.5.2017

Vedoucí práce: doc. Ing. Kamil Říha, Ph.D.

Konzultant: Ing. Pavel Dvořák, Ph.D.

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá problematikou detekce změny v digitálních obrazech, které byly pořízeny v domácích podmínkách, obyčejnou integrovanou kamerou, ve dvou po sobě jdoucích okamžicích. Jsou zde prozkoumány všechny problémy, jež tuto úlohu provází. Od předzpracování až po vyhodnocení výsledků. V průběhu práce jsou popsány a porovnány metody využívané v této oblasti, jako například rozdílová metoda (differencing) a LCP (Local Correlation Peak). Dále je zde navržena nová metoda, založená na LTP deskriptorech, která efektivně řeší stanovený problém. Tato metoda je následně podrobena testům na reálných datech. Výsledky těchto testů jsou v závěru rozebrány. Kromě metody detekce změny je v této práci navržena také metoda pro minimalizaci chyby způsobené paralaxou.

KLÍČOVÁ SLOVA

detekce změny, rozdílová metoda, LCP, LTP

ABSTRACT

This thesis concerns with change detection problematics in digital images captured under indoor conditions with an ordinary integrated camera in two consecutive moments. All challenges that accompany this problem will be discussed, starting with preprocessing and arriving to evaluation of the results. Currently used methods from this field are described and compared with each other such as differencing and LCP (Local Correlation Peak). A novel method, based on LTP descriptors, effectively solving this problem is proposed in this work. The proposed method is then tested on real data. The results of this tests are discussed subsequently. Besides the change detection method a method for parallax error minimization is proposed here.

KEYWORDS

change detection, differencing, LCP, LTP

DORAZIL, Jan *Detekce změn v digitálních obrazech*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 57 s. Vedoucí práce byl doc. Ing. Kamil Říha, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Detekce změn v digitálních obrazech“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Doc. Ing. Kamilovi Říhovi, Ph.D. a mému konzultatovi panu Ing. Pavlovi Dvořákovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	10
1 Rozbor testovaných metod	16
1.1 Registrace	16
1.1.1 Registrace na základě klíčových bodů	18
1.1.2 Metoda ECC	23
1.1.3 Vyhledávání úhlu pohledu	23
1.2 Normalizace intenzit	25
1.2.1 Globální normalizace intenzit	26
1.2.2 Lokální normalizace intenzit	27
1.3 Detekce změny	27
1.3.1 Rozdílová metoda	28
1.3.2 Poměr intenzit a metoda stínovacího modelu	28
1.3.3 Metoda LCP	29
1.3.4 Metoda LBP/LTP histogramů	30
2 Návrh a implementace řešení	35
2.1 Detekce změny	35
2.1.1 Volba metod pro detekci změny	35
2.1.2 Metoda LCP-D	40
2.1.3 Metoda LTP histogramů	40
2.1.4 Objektový návrh	41
2.2 Nalezení úhlu pohledu	43
3 Výsledky práce	45
3.1 Vyhodnocení přesnosti zvolených metod	45
3.1.1 Testovací databáze	45
3.1.2 Měření přesnosti	45
3.1.3 Vyhodnocení	47
4 Závěr	49
Literatura	50
Seznam zkratek	53
Seznam symbolů	54
Seznam příloh	56

A	Obsah elektronické přílohy	57
A.1	Složka „databáze“	57
A.2	Složka „vyhodnocení“	57
A.3	Složka „zdrojové kódy“	57

SEZNAM OBRÁZKŮ

1	Blokové schéma detekce změny	12
2	Úhel pohledu	12
3	Ukázka vstupních dat	13
4	Blokové schéma algoritmu detekce změny	14
1.1	Plocha a paralaxa	17
1.2	Problémy apertury	19
1.3	(FAST)16. pixelů rozmístěných na kružnici	20
1.4	Vliv prvků h_{31} a h_{32} na výslednou transformaci.	26
1.5	Zobecnění LBP deskriptoru.	32
1.6	Tvorba LBP deskriptoru	34
2.1	Výstup rozdílové metody	37
2.2	Výstup metody poměru intenzit	37
2.3	Výstup metody stínovacího modelu	38
2.4	Výstup metody LCP	38
2.5	Výstup metody LCP-D	39
2.6	Výstup metody LBP	39
2.7	Výstup metody LTP	40
2.8	UML diagram konečného návrhu	42
2.9	Ukázka signalizace na símku F_N	44
3.1	Závislost hodnoty f-measure na hodnotách precision a recall.	47
3.2	Porovnání přesnosti testovaných metod na testovacích databázích.	48

ÚVOD

Detekce změny v digitálních obrazech je jedním z problémů, který bývá často řešen již na nejnižších úrovních systémů počítačového vidění. Algoritmy, které obvykle provádějí tuto činnost, na základě vstupních dat (dvou či více digitálních snímků), vytvoří výstupní n -ární masku $M^{(n)}$ vyznačující umístění a druh změny, která mezi snímky nastala (viz obrázek 1). Detekce změny umožňuje např. rozlišení popředí a pozadí na záznamech pořízených digitální kamerou [2], sledování pohybu vozidel na letecky pořízených snímcích [11] a nalezení změn v pokrytí země způsobených přírodními jevy či lidskou činností [23]. Takto získané informace lze dále využít ke kompresi těchto záznamů nebo k vyvození určitých závěrů, které mohou posloužit například jako varování před přírodními katastrofami. Metody, pracující na snímcích pořízených z velké vzdálenosti od povrchu země (z letadla nebo ze satelitu), patří mezi takzvané metody dálkového průzkumu země (remote sensing). Přestože tato práce mezi metody dálkového průzkumu země nezapadá, budeme i zde používat některých pojmů známých z této oblasti.

V této práci se budeme zabývat méně prozkoumanou oblastí této problematiky, a to detekcí změny mezi snímky pořízenými ve vnitřních prostorách budov. Metoda implementovaná v této práci bude součástí většího systému určeného pro dohled nad prací servisních techniků. Samotnému používání tohoto dohledového systému bude předcházet vytvoření návodu, který bude po krocích popisovat servisní práci, jež je třeba na daném zařízení provést. V průběhu vytváření tohoto návodu bude referenční servisní technik zaznamenávat digitální snímky zařízení. Po provedení každého kroku pak bude na následujícím snímku viditelný určitý rozdíl oproti předchozímu snímku. Oblast, kde ke změně došlo, bude naší metodou označena a uložena do binární masky $M^{(2)}$. Při běžném používání dohledového systému pak nebude nutné rozlišovat veškeré změny v pořizovaných snímcích a rozhodnutí, zda byl daný krok proveden správně, bude záviset jen na předem vyznačené oblasti změny v $M^{(2)}$.

Cíle práce

Řešení této práce je rozvrženo na dvě úlohy které je třeba splnit:

- **Detekce změny.** Hlavní úlohou je návrh metody, která na snímcích detekuje pouze *strukturální změny*, tedy změny, které nastaly mechanickou manipulací se sledovaným předmětem.
- **Nalezení úhlu pohledu.** Z požadavků, které podmiňují správnou detekci změn, plyne, že vstupní snímky by měly být pořízeny ze stejného *úhlu pohledu* (tento pojem je vysvětlen na obrázku 2). Pokud tento požadavek není

splněn, detekce bude poznamenána chybami způsobenými tzv. *parallaxou* (viz sekce 1.1).

Specifikace vstupních snímků

- **Detekce změn.** Snímky, mezi nimiž budeme změnu detekovat, jsou *bitemporální* (pořízené ve dvou různých okamžicích) a *multispektrální* (v jednom okamžiku je snímáno několik vzorků z elektromagnetického kmitočtového spektra). Mezi snímky je časový odstup v řádu několika minut, během kterých dojde ke změně na sledovaném objektu. Snímky jsou snímány barevnou kamerou v modelu RGB. Jeden pixel je tudíž vyjádřen třemi vzorky z celého elektromagnetického kmitočtového spektra. Z praktických důvodů si jeden ze snímků označíme jako referenční I_1 a druhý jako zpracovávaný I_2 . Společně s těmito dvěma snímky máme dostupnou také binární masku R , která z referenčního snímku vylučuje pozadí (oblast, kde detekce změny není žádoucí). Z důvodu testování metody jsme také ručně vytvořili binární masky $G^{(2)}$, které označují oblast změny. Tyto masky považujeme za naprosto správné a podle nich budeme hodnotit výkon naší metody. Na obrázku 3 je ukázka vstupních dat, včetně zmíněných masek.
- **Nalezení úhlu pohledu.** Pro účely vyhledávání úhlu pohledu využijeme snímky

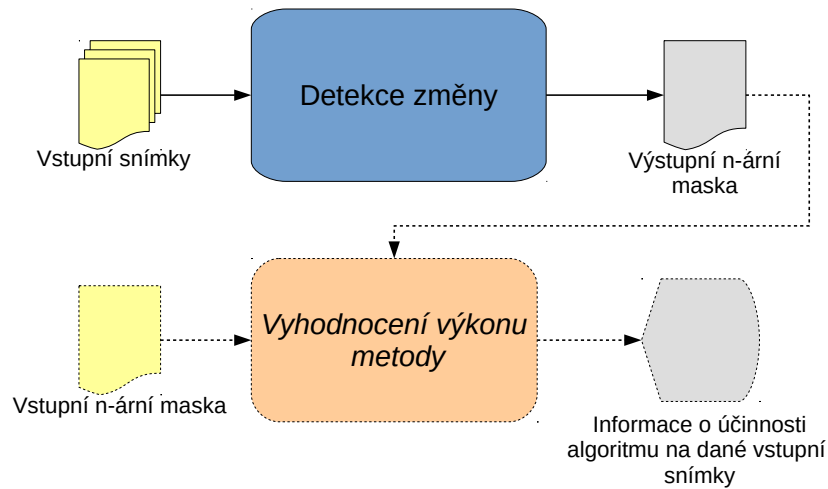
$$\{F_1, F_2, \dots, F_N\}, \quad (1)$$

získávané v reálném čase, před pořízením snímku I_2 (snímek F_N označuje aktuální snímek). Poté co uživatel nalezne správný úhel pohledu potvrdí tuto skutečnost aplikací, která pak použije snímek F_N jako I_2 pro detekci změny.

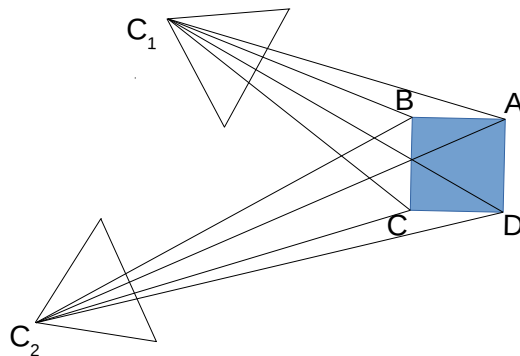
Nástin navrženého řešení

Z pohledu uživatele bude detekce změny probíhat ve dvou krocích, které odpovídají stanoveným úlohám. Tyto kroky budou provedeny v následujícím pořadí:

- **Nalezení úhlu pohledu.** Navržené řešení tohoto problému vyžaduje interakci uživatele. Na základě signalizace (popsáno dále) se požaduje, aby uživatel pohybem kamery našel správný úhel pohledu, jež odpovídá snímku I_1 . Nalezení správného úhlu uživatel potvrdí (např. stiskem tlačítka), a pořídí tak snímek I_2 .
- **Detekce změny.** Nyní je již provedena detekce změny mezi snímky I_1 a I_2 použitím jedné z navržených metod.



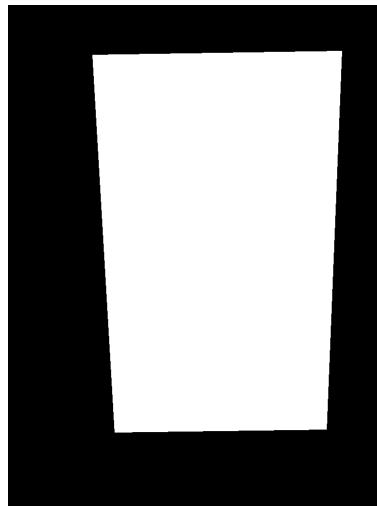
Obr. 1: Blokové schéma detekce změny. Detekční algoritmus, na základě vstupních snímků, vytvoří výstupní masku $M^{(n)}$ vyznačující umístění a druh změny. Při ladění algoritmu je pak tato maska srovnávána se vstupní n-ární maskou $G^{(n)}$ (zpravidla vytvořenou člověkem), čímž získáme informace o účinnosti použitého algoritmu na zadaných vstupních snímcích.



Obr. 2: Úhel pohledu. Na obrázku jsou zobrazeny kamery C_1 a C_2 sledující jeden objekt (čtverec ABCD) ze dvou různých úhlů pohledu. Pro dvě kamery, sledující objekt ze stejného úhlu pohledu, platí, že všechny body umístěné na objektu splňují tento vztah: $\frac{|AC_1|}{|AC_2|} = \frac{|BC_1|}{|BC_2|} = \frac{|CC_1|}{|CC_2|} = \frac{|DC_1|}{|DC_2|}$, kde $|\cdot|$ je Eukleidovská vzdálenost.



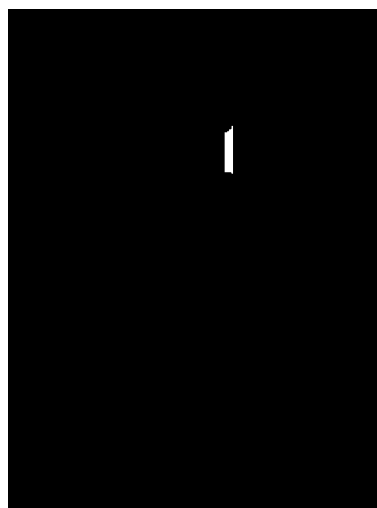
(a) Referenční snímek



(b) Maska R



(c) Zpracovávaný snímek

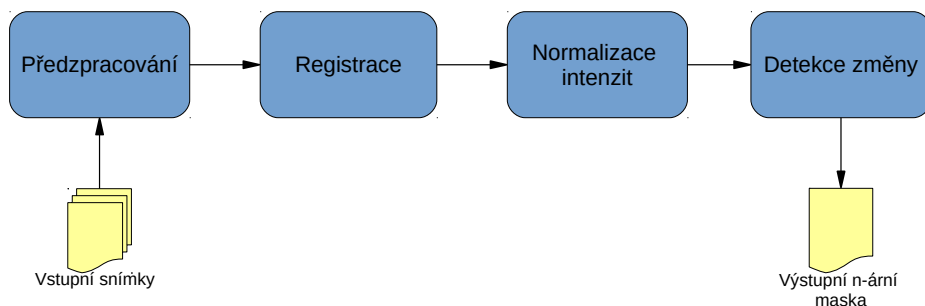


(d) Maska $G^{(2)}$

Obr. 3: Ukázka vstupních dat. V čase mezi pořízením snímku (a) a snímku (c) došlo k odebrání jedné z karet ze sledované skříně. Bílá barva v masce (b) vyčleňuje ze snímku (a) sledované popředí a černá barva ignorované pozadí snímku. Maska (d) bílou barvou vyznačuje oblast, kterou člověk považuje za odlišnou mezi snímky (a) a (c).

Krok detekce změny můžeme dále rozdělit na několik bloků, které jsou vykresleny na obrázku 4. Přesná implementace těchto bloků je popsána v kapitole 2.

1. Blok předzpracování provádí všechny operace nutné pro převedení snímků do normalizované formy. Jedná se zde například o převedení do normalizované velikosti, bitové hloubky, barevného modelu a normalizovaného rozlišení. Tento krok je nutný pro jednotnou manipulaci se snímky v dalších blocích.
2. Blok registrace zajišťuje prostorové zarovnání vstupních snímků. Pokud bychom takto zpracované snímky položili na sebe, všechny hrany sledovaného předmětu by se pak překrývaly. Podle charakteru snímků může být registrace provedena jednoduchou *translací*, *affinní transformací* nebo dokonce *perspektivní transformací*. Registrace obvykle nebývá dokonalá a některé vlivy perspektivní deformace touto metodou nelze odstranit. Je proto důležité v následujících blocích počítat s jistou chybou registrace.
3. Blok normalizace intenzit kompenzuje změny osvětlení, času expozice a dalších jevů, které ovlivňují světlost snímků. U barevných snímků je třeba normalizovat každý barevný kanál samostatně¹. Do normalizace intenzit je možné zahrnout také kompenzace lokálních změn v osvětlení (např. stíny).
4. Blok detekce změny se musí dokázat vyrovnat se všemi vlivy, které nebyly odstraněny předchozími bloky. Patří mezi ně šum kamery, chyby registrace, perspektivní deformace, zbylé rozdíly v intenzitách, chyby zaostření kamery, rozostření pohybem, chyby způsobené omezeným rozlišením kamery a jiné.



Obr. 4: Blokové schéma algoritmu detekce změny. Vstupní snímky jsou předány bloku předzpracování. Výstupem bloku detekce změny je již maska $M^{(n)}$.

¹Změna ve spektru světelného zdroje způsobí rozdílné poměry intenzit mezi barevnými kanály. Pokud tedy nelze zajistit jednotný zdroj osvětlení pro všechny snímky, není možné normalizovat všechny kanály společně.

Současný stav problematiky

Jak již bylo řečeno v úvodu, současné metody detekce změny se zabývají převážně detekcí změny mezi snímky pořízenými statickou kamerou. Nejčastější aplikací těchto metod je implementace tzv. *background-subtraction* systému, který statické oblasti (neměnicí se) označuje jako pozadí (background) a měnící se oblasti jako popředí (foreground). Tyto metody jsou obvykle založeny na statistickém modelování popředí a pozadí na základě sekvence vstupních snímků. Dle těchto modelů je následně rozhodnuto (např. statistickým testováním), do které kategorie náleží zkoumaná oblast. Výhodou tohoto přístupu je možnost aktualizace vytvořených statistických modelů za běhu, čímž je možné kompenzovat např. vlivy osvětlení, které se v průběhu snímání scény mění. Na tomto principu je založena také aktuálně nejvýkonnější metoda [24], která využívá pro modelování popředí a pozadí konvoluční neuronové sítě (convolution neural network). Uvedená metoda dosahuje na testovacích databázích hodnotu f-measure (viz sekce 3.1.2) 0,95, což se již blíží schopnostem lidského oka [24]. Další moderní metoda založená na tomto principu je uvedena v práci [21]. Pro vytvoření statistického modelu pozadí jsou zde použity texturální příznaky, které samy dokážou potlačit vliv osvětlení (model popředí zde není známý).

Metod, které se zabývají detekcí změny mezi snímky pořízenými pohyblivou kamerou, existuje o poznání méně. Jelikož neexistuje žádný projekt, který by se zabýval vyhodnocením těchto metod, je také problematické rozhodnout, které metody vypovídají o současném stavu problematiky. Jedna z takových metod je navržena v disertační práci [3]. Tato metoda je založena na principu, který je popsán v sekci 1.3.3, a umožňuje správnou detekci změny za přítomnosti chyby registrace a chyby způsobené paralaxou (viz sekce 1.1). Jelikož snímky zpracovávány v uvedené práci jsou pořízeny z velké výšky nad zemí, chyba způsobena paralaxou je minimální. Hodnota f-measure, dosažená v této práci, se pohybuje kolem 0,85. Další prací která se zabývá podobnou problematikou je [14]. Tato práce nabízí řešení detekce nově přítomného předmětu v sekvenci snímků, pořízených kamerou na pojízdném vozidle. Na snímcích, zarovnaných pomocí homografie (viz sekce 1.1), je v této práci provedena detekce změny rozdílovou metodou 1.3.1 (po filtraci pomocí Sobelova operátoru).

1 ROZBOR TESTOVANÝCH METOD

1.1 Registrace

Jak již bylo řečeno v předchozí kapitole, registrace zajišťuje prostorové zarovnání vstupních snímků. Registrační metody obvykle pracují na dvou snímcích, kde jeden je referenční I_1 a druhý zpracováváný I_2 , což odpovídá našemu zavedenému značení. Samotnou registraci provedeme maticovou transformací snímku I_2 , jejíž parametry je třeba určit. Volba použité maticové transformace závisí na podmínkách, za kterých jsou snímky pořízeny.

Perspektivní projekce a ztráta projection depth. Jelikož snímky, se kterými pracujeme, jsou pořízeny v reálném světě, mezi skutečnou scénou a dvourozměrným snímkem z kamery existuje vztah daný perspektivní projekcí. V *homogenním souřadném systému* [22] platí mezi trojrozměrným bodem $\tilde{\mathbf{p}} \in \mathbb{P}^3$, $\mathbb{P}^3 = (\mathbb{R}^4 \setminus \{\mathbf{0}\})$ a jeho dvourozměrnou projekcí $\tilde{\mathbf{x}} \in \mathbb{P}^2$, $\mathbb{P}^2 = (\mathbb{R}^3 \setminus \{\mathbf{0}\})$ vztah [22]:

$$\tilde{\mathbf{x}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{\mathbf{p}}. \quad (1.1)$$

Převedením *homogenního* vektoru $\tilde{\mathbf{x}}$ na nehomogenní,

$$\mathbf{x} = \begin{pmatrix} \frac{x_{\tilde{\mathbf{x}}}}{w_{\tilde{\mathbf{x}}}} & \frac{y_{\tilde{\mathbf{x}}}}{w_{\tilde{\mathbf{x}}}} \end{pmatrix}^T, \tilde{\mathbf{x}} = \begin{pmatrix} x_{\tilde{\mathbf{x}}} & y_{\tilde{\mathbf{x}}} & w_{\tilde{\mathbf{x}}} \end{pmatrix}^T, \quad (1.2)$$

získáme výsledné souřadnice dvourozměrné projekce bodu $\tilde{\mathbf{p}}$. Při převodu z homogenního souřadného systému došlo k zahození složky $w_{\tilde{\mathbf{x}}}$, čímž jsme přišli o informaci o vzdálenosti bodu $\tilde{\mathbf{p}}$ od projekční plochy (tzv. *projection depth*). Ze znalosti vektoru \mathbf{x} již není možné získat původní vektor $\tilde{\mathbf{p}}$. Z toho plyne, že není možné z dvourozměrného snímku rekonstruovat původní trojrozměrnou scénu.

Mapování snímku z jedné kamery na druhou. Uvažujme nyní situaci, kdy dvě kamery (C_1, C_2) snímají jednu scénu. Známe jejich polohu \mathbf{t}_1 a \mathbf{t}_2 , jejich rotaci \mathbf{R}_1 a \mathbf{R}_2 a zároveň *kalibrační matici* (calibration matrix [22]) \mathbf{K}_1 a \mathbf{K}_2 . Z těchto parametrů určíme tzv. *matici kamery* (camera matrix [22]) $\tilde{\mathbf{P}}_1$ a $\tilde{\mathbf{P}}_2$, kde

$$\tilde{\mathbf{P}}_i = \begin{pmatrix} \mathbf{K}_i & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_i & \mathbf{t}_i \\ \mathbf{0}^T & 1 \end{pmatrix} = \tilde{\mathbf{K}}_i \mathbf{E}_i. \quad (1.3)$$

Snímek z kamery C_1 nyní představuje I_1 a snímek z kamery C_2 představuje I_2 . Pokud bychom znali také projection depth bodu na poloze $\tilde{\mathbf{x}}^{(2)}$ ve snímku I_2 , bylo

by možné určit zpětnou transformací polohu \mathbf{p} tohoto bodu v původní trojrozměrné scéně.

$$\mathbf{p} \sim \tilde{\mathbf{E}}_2^{-1} \tilde{\mathbf{K}}_2^{-1} \tilde{\mathbf{x}}^{(2)}. \quad (1.4)$$

Projekcí bodu \mathbf{p} pak můžeme určit polohu $\tilde{\mathbf{x}}^{(1)}$ tohoto bodu ve snímku I_1 .

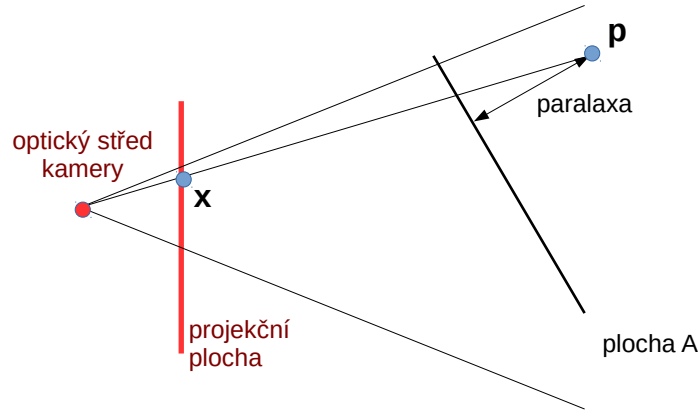
$$\tilde{\mathbf{x}}^{(1)} \sim \tilde{\mathbf{K}}_1 \tilde{\mathbf{E}}_1 \mathbf{p} \quad (1.5)$$

Kombinací těchto transformací lze přímo z $\tilde{\mathbf{x}}^{(2)}$ určit $\tilde{\mathbf{x}}^{(1)}$ sledovaného bodu.

$$\tilde{\mathbf{x}}^{(1)} \sim \tilde{\mathbf{P}}_1 \tilde{\mathbf{P}}_2^{-1} \tilde{\mathbf{x}}^{(2)} = \mathbf{M}_{12} \tilde{\mathbf{x}}_0 \quad (1.6)$$

Při snímání obyčejnou kamerou však obvykle *projection depth* neznáme, tudíž v obecném případě tuto transformaci nelze použít.

Homografie a paralaxa. Budeme-li předpokládat, že každý trojrozměrný bod sledované scény leží na ploše $A : \mathbf{n}_0 \mathbf{p} + c_0 = 0$, je možné zahodit poslední sloupec matice \mathbf{M}_{12} (vzorec 1.6) [22]. Jelikož nás nezajímá výsledná *projection depth*, můžeme zahodit také poslední řádek této matice. Touto úpravou dostaneme standartní 3x3 matici $\tilde{\mathbf{H}}_{12}$ dvourozměrné perspektivní transformace (tzv. *homografie* [homography]). Body ležící mimo plochu A způsobují chybu v registraci. Vzdálenost bodu od plochy A se často nazývá paralaxa ¹ (obr. 1.1).



Obr. 1.1: Paralaxa – vzdálenost bodu od plochy A.

Jelikož na snímcích, které zpracováváme často dominuje rovná plocha, je právě použití homografie opodstatněné. S chybou registrace způsobenou paralaxou je však nutné počítat a její vlivy kompenzovat.

¹V některé literatuře paralaxa představuje úhel, v této práci však vycházíme z terminologie v knize [22].

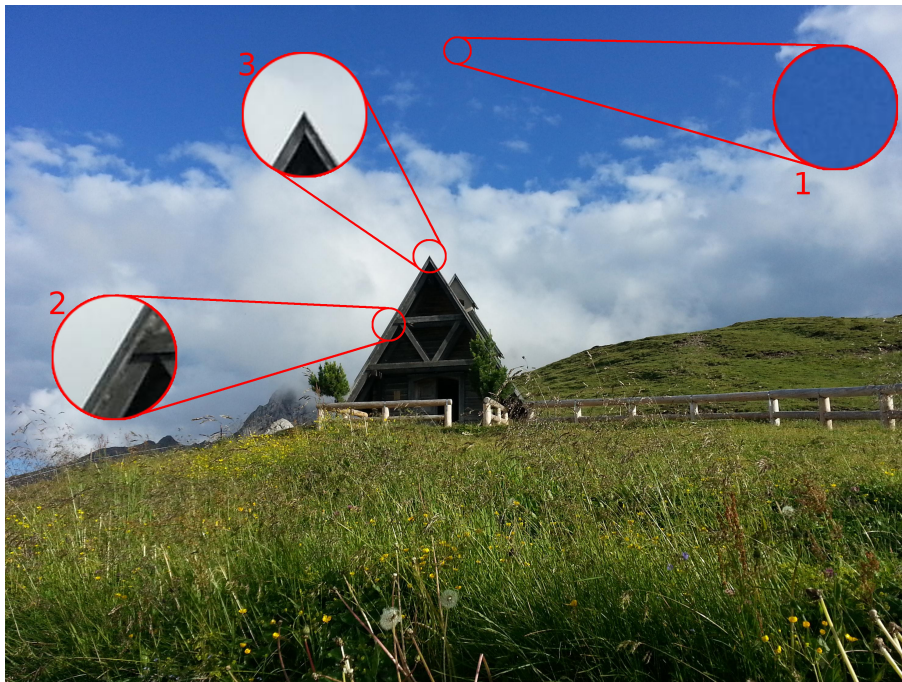
1.1.1 Registrace na základě klíčových bodů

Registrační metody na základě *klíčových bodů* (keypoints) se obvykle skládají z těchto kroků:

- Detekce klíčových bodů na obou registrovaných snímcích (feature detection).
- Popis detekovaných bodů na základě okolních pixelů (feature description).
- Párování těchto bodů (k bodu ze snímku I_1 je přiřazen odpovídající bod ze snímku I_2) podle popisu získaného v předchozím kroku (feature matching).
- Výpočet transformační matice na základě rozdílů v poloze spárovaných bodů.
- Transformace snímku I_2 .

Klíčové body. Klíčové body jsou body, které lze snadno lokalizovat na různorodých snímcích snímaného předmětu. Tyto body jsou pevně svázány s jejich okolními pixely, které budeme nazývat výstřižky (patches). Bez tohoto výstřižku není možné lokalizovat klíčový bod ve snímku. Výstřižky často zachycují vrcholky hor či rohy budov. Dobré výstřižky jsou co nejméně poznamenány tzv. *problémy apertury* (aperture problems). Problém apertury způsobuje nejistotu při detekci přesné polohy klíčového bodu v jednotlivých snímcích. Na obrázku 1.2 vidíme tři výstřižky vybrané ze zobrazeného snímku. Výstřižek 1 obsahuje pouze hladkou modrou plochu - není možné s jistotou určit jeho polohu. Umístění výstřižku 2 bychom odhadli někde na hraně střechy, ovšem není možné jej určit přesně. Předchozí dva výstřižky jsou poznamenány problémy apertury. Poslední, výstřižek 3, je pro umístění klíčového bodu nejvhodnější. Polohu tohoto výstřižku ve snímku můžeme určit velice přesně.

Deskriptory. Popisem klíčových bodů rozumíme proces, kdy výstřižek okolo klíčového bodu transformujeme do formy *deskriptoru*. Na základě podobnosti deskriptorů budeme později rozhodovat, které klíčové body označují stejnou oblast, což nám umožní jejich párování.



Obr. 1.2: Problémy apertury. Výstřižky 1 a 2 jsou pro umístění klíčového bodu nevhodné. Výstřižek 3 je pro umístění tohoto bodu vhodný - jeho polohu je možné jednoznačně určit.

Volba metody pro detekci a popis klíčových bodů

Pro detekci klíčových bodů a jejich popis existuje mnoho metod. Neznámějšími jsou SIFT (Scale-Invariant Feature Transform) a SURF (Speeded-Up Robust Features). Jelikož uvedené metody jsou patentovány, je pro komerční použití vhodné zvolit jinou metodu. Z důvodu snadno dostupné implementace v OpenCV, jsme v této práci zvolili metodu ORB [18] (Oriented BRIEF).

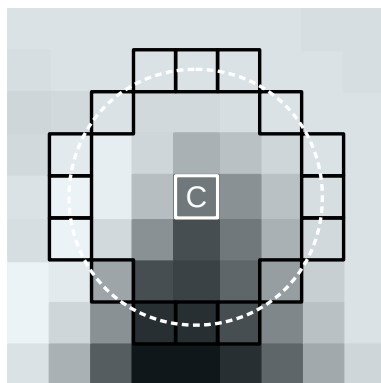
ORB je metoda pro detekci a popis příznaků vycházející z algoritmů FAST (Features from Accelerated Segment Test) a BRIEF (Binary Robust Independent Elementary Features). V následujících odstavcích budou proto krátce popsány i tyto algoritmy.

Algoritmus FAST

Algoritmus FAST [17] vychází z tzv. *testu segmentu*, který na základě 16. okolních pixelů bodu c na poloze \mathbf{x}_c (rozmístěných na kružnici okolo tohoto bodu, viz obr. 1.3) rozhoduje, zda tento bod na snímku označuje polohu rohu (tedy reprezentuje možný klíčový bod). Uvažujme snímek daný funkcí $i = I(\mathbf{x})$, která vektoru polohy \mathbf{x} přiřazuje intenzitu i . Pokud okolo bodu c existuje souvislá sekvence n pixelů s intenzitou

větší než $I(\mathbf{x}_c) + t$, na poloze \mathbf{x}_c je detekován klíčový bod (t je volitelná hodnota prahu).

Použitím algoritmu ID3 (Iterative Dichotomiser 3) je na trénovací databázi snímků vytvořen *rozhodovací strom* (decision tree), který efektivně implementuje testování segmentu. Vytvořený rozhodovací strom je nakonec převeden do jazyka C a zkompilován. Tím získáme funkci pro testování segmentu, která je použita v konečné implementaci algoritmu FAST. Je vhodné poznamenat, že výsledky získané touto implementací nemusí ve skutečnosti odpovídat přímému výpočtu testu segmentu. Rozhodovací strom je specializovaný na typy rohů, které se vyskytují v databázi trénovacích snímků.



Obr. 1.3: (FAST) 16. pixelů rozmístěných na kružnici. Pokud má n sousedících pixelů intenzitu větší než $I(\mathbf{x}_c) + t$, bod c je označen jako klíčový. Vhodná volba hodnoty n je okolo 12 (detekuje rohy pod úhlem 45°).

Algoritmus BRIEF

Nejznámější metody pro tvorbu deskriptorů jako SIFT a SURF extrahují z výstřižku okolo klíčového bodu vektory 128 resp. 64 hodnot s plovoucí řádovou čárkou [4]. Porovnávání těchto vektorů je relativně náročné na výpočetní výkon. Z tohoto důvodu některé metody provádějí na těchto vektorech redukci dimensionalit pomocí PCA [10] (Principal Component Analysis) či LDA [7] (Linear Discriminant Analysis), případně dokonce redukci na binární řetězec [19] (např. pomocí *hašovací funkce*). Poslední uvedený přístup, tedy redukce na binární řetězec, umožňuje porovnávání deskriptorů *Hammingovou vzdáleností*, jejíž výpočet je na dnešních počítačích velmi rychlý.

BRIEF [4] vytváří deskriptory v podobě binárního řetězce již od počátku. Redukce těchto dat tedy není nutná a výsledné deskriptory je možné rychle porovnávat použitím Hammingovy vzdálenosti. Jednotlivé bity deskriptoru jsou v algoritmu

BRIEF určeny jednoduchým porovnáním dvou pixelů (na poloze \mathbf{x}_1 a \mathbf{x}_2) z výstřižku o velikosti $S \times S$. Pokud platí $I(\mathbf{x}_1) < I(\mathbf{x}_2)$, bit bude nabývat hodnoty 1, v opačném případě 0. Polohy srovnávaných pixelů jsou získány vzorkováním z normálního rozdělení se střední hodnotou $\mu = 0$ a rozptylem $\sigma^2 = \frac{1}{25}S^2$. Počet porovnávaných párů pixelů (n_d) se může mezi různými implementacemi algoritmu lišit. Ze zavedeného značení BRIEF- k lze n_d určit jako $n_d = 8 \cdot k$.

Metoda ORB

Oblast použití algoritmu FAST je značně zúžená, protože poloha a množství detekovaných klíčových bodů je velmi závislá na měřítku snímku. Algoritmus také neposkytuje informaci o orientaci detekovaných rohů, která je často vyžadována. Problémem s algoritmem BRIEF je zase závislost jeho výkonu na rotaci snímku. Metoda ORB řeší všechny tyto problémy [18].

Rozdílná měřítka snímků. Pro řešení problému s měřítkem snímku u algoritmu FAST využívá ORB tzv. *měřítkovou pyramidu* (scale pyramid). Měřítko původního snímku je postupně zvětšováno a zmenšováno, čímž získáme množství snímku o různých velikostech. Seřazením těchto snímku nad sebe podle velikosti dostaneme tvar pyramidu. Pomocí FAST poté detekujeme klíčové body na všech vrstvách pyramidu. Klíčové body jsou následně ohodnoceny pomocí *Harrisova měřítka* (Harris measure), čímž získáme informaci o jejich kvalitě. Z těch nejkvalitnějších je nakonec vybrán požadovaný počet klíčových bodů.

Orientace rohů. Pro získání informace o orientaci rohů využívá ORB tzv. *těžiště intenzity* (intensity centroid). Definujme si nyní oblast výstřižku jako množinu $\mathcal{V} \in \mathbb{R}^2$, která obsahuje polohové vektory všech pixelů z výstřižku. Pro tuto oblast určíme vektor udávající orientaci rohu jako

$$\mathbf{C} = \begin{pmatrix} m_x & m_y \\ \bar{I}_{\mathcal{V}} & \bar{I}_{\mathcal{V}} \end{pmatrix}^T, \quad (1.7)$$

kde $\bar{I}_{\mathcal{V}} = \sum_{\mathbf{x} \in \mathcal{V}} I(\mathbf{x})$ je průměrná intenzita výstřižku a

$$m_x = \sum_{\mathbf{x}=(x,y) \in \mathcal{V}} x I(\mathbf{x}), \quad (1.8)$$

$$m_y = \sum_{\mathbf{x}=(x,y) \in \mathcal{V}} y I(\mathbf{x}) \quad (1.9)$$

jsou momenty na ose x resp. y.

Závislost na rotaci. Hodnota deskriptorů získaných algoritmem *BRIEF* je značně závislá na rotaci snímků [18], což způsobuje chyby při párování klíčových bodů. ORB pro řešení tohoto problému využívá informaci o orientaci získané v předchozím kroku (viz odstavec Orientace rohů). Z vektoru \mathbf{C} je nejprve určen úhel $\theta = \text{atan2}(m_y, m_x)$, kde $\text{atan2}()$ odpovídá funkci arkus tangents, která určí správný úhel vektoru k ose x ve všech možných kvadrantech. Pomocí rotační matice

$$\mathbf{R}_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (1.10)$$

je pak matice párů pixelů $x_1^{(i)}, x_2^{(i)}$ vybraných pro srovnání transformována:

$$\mathbf{S}_\theta = \mathbf{R}_\theta \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(n)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(n)} \end{pmatrix}. \quad (1.11)$$

Algoritmus BRIEF takto prakticky určuje deskriptor na výstřižku, který je již orientován podle rohu, jež zobrazuje. Tento přístup ovšem přináší další problém, který je nutné řešit [18].

Párování klíčových bodů

Pro párování klíčových bodů jsme v této práci zvolili metodu hrubé síly kombinovanou s testem poměru vzdáleností [9]. Uvažujme nyní, že máme dostupné deskriptory $\{\mathbf{d}_i^{(1)}\}_{i=1}^I$ a $\{\mathbf{d}_j^{(2)}\}_{j=1}^J$ klíčových bodů snímku I_1 resp. I_2 . Porovnáním každého deskriptoru snímku I_1 s každým deskriptorem snímku I_2 pomocí metriky $m_{ij} = m(\mathbf{d}_i^{(1)}, \mathbf{d}_j^{(2)})$ získáme matici vzdáleností těchto deskriptorů.

$$\mathbf{D}_{12} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1J} \\ m_{21} & m_{22} & \dots & m_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ m_{I1} & m_{I2} & \dots & m_{IJ} \end{pmatrix} \quad (1.12)$$

Nalezením dvou nejmenších hodnot na řádku i této matice pak dostaneme dva deskriptory $\mathbf{d}_k^{(2)}, \mathbf{d}_l^{(2)}$, které jsou potencionálně nejvhodnějším protějškem k deskriptoru $\mathbf{d}_i^{(1)}$. Pokud vzdálenosti těchto deskriptorů k deskriptoru $\mathbf{d}_i^{(1)}$ splňují podmínku:

$$m_{ik} < \gamma \cdot m_{il}, \quad (1.13)$$

pak vytvoříme pár $(\mathbf{d}_i^{(1)}, \mathbf{d}_k^{(2)})$ a zároveň také spárujeme odpovídající klíčové body i a k . Tuto operaci opakujeme pro každý řádek matice.

Jako metriku m jsme v této práci použili *Hammingovu vzdálenost*, která je pro porovnávání binárních deskriptorů metody ORB nejvhodnější. Volba konstanty γ ovlivňuje poměr správně přiřazených klíčových bodů k chybně přiřazeným klíčovým bodům. V původním dokumentu [9] autor použil hodnotu $\gamma = 0,8$.

1.1.2 Metoda ECC

ECC (Enhanced Correlation Coefficient maximization) je metoda používaná pro optimalizaci transformační matice určené k registraci snímků.

V našem případě budeme metodu používat pro optimalizaci matice $\tilde{\mathbf{H}}_{12}$. Zavedme si nyní vektor $\mathbf{h} = (h_{00} \ h_{10} \ \dots \ h_{22})^T$ obsahující všechny prvky matice $\tilde{\mathbf{H}}_{12}$. Transformace $\phi(\mathbf{x}, \mathbf{h})$ na základě vektoru parametrů \mathbf{h} transformuje vektor souřadnic \mathbf{x} (tedy mapuje body z I_1 do I_2) pomocí homografie. Dále vektory

$$\mathbf{i}_r = \begin{bmatrix} \mathbf{I}_{REF}(\mathbf{x}_1) & \mathbf{I}_{REF}(\mathbf{x}_2) & \dots & \mathbf{I}_{REF}(\mathbf{x}_K) \end{bmatrix}^T, \quad (1.14)$$

$$\mathbf{i}_p(\mathbf{h}) = \begin{bmatrix} \mathbf{I}_{PR}(\phi(\mathbf{x}_1, \mathbf{h})) & \mathbf{I}_{PR}(\phi(\mathbf{x}_2, \mathbf{h})) & \dots & \mathbf{I}_{PR}(\phi(\mathbf{x}_K, \mathbf{h})) \end{bmatrix}^T \quad (1.15)$$

obsahují intenzity všech $(1 \dots K)$ pixelů snímku I_1 a k nim náležící pixely z I_2 (náležitost je dána transformací $\phi(\mathbf{x}, \mathbf{h})$). Odečtením aritmetického průměru od těchto vektorů získáme jejich normalizované formy $\bar{\mathbf{i}}_r$ a $\bar{\mathbf{i}}_p$.

Optimálních hodnot parametrů \mathbf{h} je dosaženo maximalizací *vylepšeného korelačního koeficientu* [5] (Enhanced Correlation Coefficient):

$$r(\mathbf{h}) = \frac{\bar{\mathbf{i}}_r^T \bar{\mathbf{i}}_p(\mathbf{h})}{\|\bar{\mathbf{i}}_r\| \|\bar{\mathbf{i}}_p(\mathbf{h})\|}, \quad (1.16)$$

kde $\|\cdot\|$ je Eukleidovská norma. Tento koeficient je nezávislý na změnách v osvětlení a výsledný problém lze reprezentovat jako soustavu lineárních rovnic [5]. Soustava lineárních rovnic je v této metodě řešena iteračně.

Na rozdíl od metod založených na detekci příznaků, tato metoda pracuje se všemi pixely snímků – je tedy velice robustní. Její nevýhodou je nutná znalost počátečního odhadu matice $\tilde{\mathbf{H}}_{12}$ před započítáním výpočtu. Pokud je odhad nepřesný, metoda nemusí konvergovat ani v mnoha iteracích. Tento problém je možné řešit rozdělením registrace na dvě části. Nejprve je metodou ECC na podvzorkovaných snímcích určen odhad $\tilde{\mathbf{H}}_{12}$, který je v druhém kroku upřesněn výpočtem na původních snímcích. Pokud je mezi snímky I_1 a I_2 jen malý posun, vhodným počátečním odhadem je jednotková matice.

1.1.3 Vyhledávání úhlu pohledu

Navržená metoda pro řešení tohoto problému vychází z prvků matice²

$$\tilde{\mathbf{H}}_N = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}, \quad (1.17)$$

²Jednička na třetím řádku, ve třetím sloupci matice doplňuje matici do homogenního tvaru [22].

což je matice homografie vypočtená mezi I_1 a F_N . Podle [6] lze matici homografie rozložit na součin několika matic:

$$\tilde{\mathbf{H}}_N = \tilde{\mathbf{H}}_N^s \tilde{\mathbf{H}}_N^a \tilde{\mathbf{H}}_N^p = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & 1 \end{pmatrix}. \quad (1.18)$$

Matice $\tilde{\mathbf{H}}_N^s$ je maticí *podobnostní transformace* (similarity transform [22]). Matice $\tilde{\mathbf{H}}_N^a$ je maticí speciálního případu *afinní transformace* [22] a konečně $\tilde{\mathbf{H}}_N^p$ je maticí speciálního případu *projekční transformace* [22]. Jednoduchým vynásobením těchto matic lze dokázat, že vektor $\mathbf{v} = (h_{31} \ h_{32})^T$. Použijeme-li rozklad ze vzorce 1.18 pro transformaci vektoru $\tilde{\mathbf{x}}$, dostaneme:

$$\tilde{\mathbf{x}}' = \tilde{\mathbf{H}}_N \tilde{\mathbf{x}} = \tilde{\mathbf{H}}_{1C}^s \tilde{\mathbf{H}}_N^a \tilde{\mathbf{H}}_N^p \tilde{\mathbf{x}}, \quad (1.19)$$

což lze reprezentovat jako posloupnost tří po sobě následujících transformací:

$$\tilde{\mathbf{x}}' = \tilde{\mathbf{H}}_N^s [\tilde{\mathbf{H}}_N^a (\tilde{\mathbf{H}}_N^p \tilde{\mathbf{x}})]. \quad (1.20)$$

Projekční transformace je tedy provedena dříve než podobnostní transformace (tato informace bude využita dále).

Uvažujeme nyní čtverec daný body $A = (-1 \ -1)^T$, $B = (-1 \ 1)^T$, $C = (1 \ 1)^T$, $D = (1 \ -1)^T$. Zobrazení tohoto čtverce v závislosti na prvcích matice $\tilde{\mathbf{H}}_N^p$ získáme zobrazení uvedené na obrázcích 1.4. Z těchto obrázků lze vypočítat, že pro $h_{31} = 0, h_{32} = 0$ bude úhel pohledu I_1 odpovídat úhlu pohledu I_2 . Pokud úhel pohledu obrázku I_1 odpovídá úhlu pohledu I_2 , chyba detekce změny způsobená paralaxou bude nulová.

Signalizace odchylky úhlu pohledu. Vektoru \mathbf{v} je dokonce možné využít pro signalizaci míry odchylky od požadovaného úhlu pohledu. Prvek h_{31} signalizuje odchylku na ose x a prvek h_{32} odchylku na ose y. Problém však nastává, pokud mezi pořízením snímků I_1 a F_N dojde k rotaci kamery kolem osy z. Označme si nyní souřadný systém před podobnostní transformací jako S a souřadný systém po podobnostní transformaci jako S' . Jelikož je podobnostní transformace při transformaci maticí $\tilde{\mathbf{H}}_N$ provedena až po projekční transformaci, vektor \mathbf{v} vypovídá o transformaci v souřadném systému S . Pokud matice $\tilde{\mathbf{H}}_N$ reprezentuje transformaci $I_1 \rightarrow F_N$ je třeba vektor \mathbf{v} rotovat, aby signalizace odpovídala souřadnému systému S' (tedy aby bylo možné signalizaci zobrazit ve snímku F_N). Z tohoto důvodu potřebujeme z matice $\tilde{\mathbf{H}}_N$ určit rotační matici, kterou podobnostní transformace rotuje souřadný systém S . Ze vzorce 1.18 plyne, že pro submatici $\mathbf{H}_{N,33}$ platí:

$$\mathbf{H}_{N,33} = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} = s\mathbf{R}\mathbf{K} + \mathbf{t}\mathbf{v}^T. \quad (1.21)$$

Jelikož vektory \mathbf{t} a \mathbf{v} lze určit přímo z matice \mathbf{H}_N , jako: $\mathbf{t} = (h_{13} \ h_{23})^T$ a $\mathbf{v} = (h_{31} \ h_{32})$, není problém jejich násobek od matice $\mathbf{H}_{N,33}$ odečíst.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \mathbf{H}_{N,33} - \mathbf{t}\mathbf{v}^T = {}_s\mathbf{R}\mathbf{K} \quad (1.22)$$

Díky tomu, že rotační matice

$$\mathbf{R} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (1.23)$$

je ortogonální ($\mathbf{R}\mathbf{R}^T = \mathbf{I}$), lze na matici \mathbf{A} použít *QR rozklad*. Použitím Gram-Schmidtova procesu získáme:

$$\mathbf{R} = (\mathbf{u}_1 \ \mathbf{u}_2), \quad (1.24)$$

kde $\mathbf{u}_1 = \frac{1}{\sqrt{a_{11}^2 + a_{21}^2}}(a_{11} \ a_{21})^T$. Hodnotu vektoru \mathbf{u}_2 již nepotřebujeme počítat. Dle vzorce 1.23 je zřejmé, že hledaná rotační matice bude:

$$\mathbf{R} = \frac{1}{\sqrt{a_{11}^2 + a_{21}^2}} \begin{pmatrix} a_{11} & -a_{21} \\ a_{21} & a_{11} \end{pmatrix}. \quad (1.25)$$

Rotací vektoru \mathbf{v} nyní získáme vektor \mathbf{v}' , který lze libovolným způsobem vykreslit na snímku F_N jako signalizaci odchylky úhlu pohledu (např. tak jak je uvedeno na obrázcích 1.4).

$$\mathbf{v}' = \mathbf{R}\mathbf{v} \quad (1.26)$$

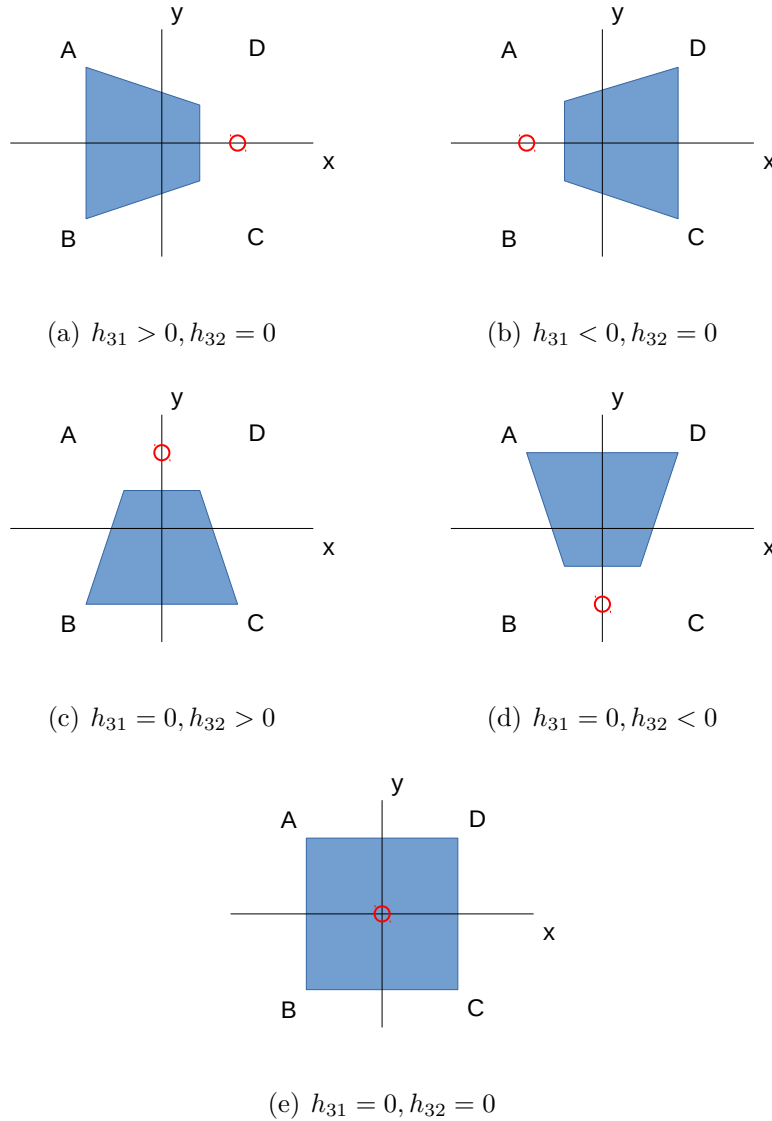
Na základě této signalizace pak bude nutné, aby uživatel našel úhel pohledu, jež odpovídá I_1 .

1.2 Normalizace intenzit

Normalizace intenzit kompenzuje změny osvětlení, doby expozice a dalších jevů, které nastanou mezi pořízením snímků I_1 a I_2 , a ovlivňují světlost snímků. Pro úspěšnou detekci změn je tento krok velice důležitý. Některé metody pro detekci změny však provádějí normalizaci intenzit samy, tudíž by bylo možné tento krok za určitých okolností vynechat.

Volba metody pro normalizaci intenzit je závislá na jevech, které je třeba kompenzovat. Nejdůležitější z těchto jevů jsou:

- doba expozice,
- velikost apertury clony kamery,
- zesílení signálu ze snímače,
- osvětlení scény.



Obr. 1.4: Vliv prvků h_{31} a h_{32} na výsledek transformace. Ostatní prvky matice $\tilde{\mathbf{H}}_N$ odpovídají jednotkové matici 3×3 . Červené kružnice signalizují odchylku \mathbf{v}' .

1.2.1 Globální normalizace intenzit

Kompenzace prvních tří jevů (viz sekce 1.2) je možná globální úpravou všech intenzit pixelů dle vzorce:

$$I'(\mathbf{x}_i) = s \cdot (I(\mathbf{x}_i) + o), \quad (1.27)$$

pro všechna $\mathbf{x}_i \in \mathcal{I}$. (oblast $\mathcal{I} \in \mathbb{R}^2$ je množina polohových vektorů všech pixelů snímku.) Hodnoty o a s jsou předem určeny z intenzit $I(\mathbf{x}_i)$.

Normalizace průměrné hodnoty a rozptylu. Tato metoda [15] provádí globální normalizaci intenzit podle vzorce 1.27. Hodnota o je zde určena jako opačná

hodnota aritmetického průměru intenzit pixelů na snímku ($o = -\mu$) a s odpovídá převrácené hodnotě rozptylu intenzit ($s = \frac{1}{\sigma^2}$). Výsledkem normalizace je pak snímek, jehož pixely mají průměrnou intenzitu rovnu nule a rozptyl intenzit roven jedné.

1.2.2 Lokální normalizace intenzit

Kompenzace čtvrtého uvedeného jevu (viz sekce 1.2), tedy změny v osvětlení scény, již touto úpravou řešit nelze. S použitím zjednodušeného modelu odrazivosti světla, kde uvažujeme pouze *diffuzní odrazivost* (diffuse reflection), lze intenzitu odraženého paprsku vyjádřit jako [22]:

$$L_d(\lambda) = L_i(\lambda)f_d(\lambda)[\mathbf{v}_i \cdot \mathbf{n}]^+, \quad (1.28)$$

kde

$$[\mathbf{v}_i \cdot \mathbf{n}]^+ = \max(0, \mathbf{v}_i \cdot \mathbf{n}), \quad (1.29)$$

L_i je intenzita dopadajícího světelného paprsku ve směru \mathbf{v}_i , \mathbf{n} je normálový vektor plochy, na kterou paprsek dopadá a $f_d(\lambda)$ je BRDF (Bidirectional Reflectance Distribution Function) této plochy pro vlnovou délku λ . Povšimněme si, že v tomto modelu množství vyzařeného světla nezávisí na směru, ze kterého plochu sledujeme. Určením BRDF pro každý pixel bychom získali snímek, který je nezávislý na osvětlení. Jelikož obvykle neznáme směr a intenzitu dopadajícího paprsku, je určení BRDF prakticky nemožné.

Homomorfní filtr. Převědeme-li intenzitu pixelů snímku podle vzorce $I'(\mathbf{x}) = \log(I(\mathbf{x}))$, intenzita odraženého paprsku bude nyní dána součtem

$$L_d(\lambda) = L_i(\lambda) + f_d(\lambda) + [\mathbf{v}_i \cdot \mathbf{n}]^+. \quad (1.30)$$

Předpokládejme, že hodnota $L_i(\lambda) + [\mathbf{v}_i \cdot \mathbf{n}]^+$ je dána polohou zdroje světla a intenzitou dopadajícího světla. Složka $f_d(\lambda)$ je dána pouze odrazivostí sledované plochy. Intenzita dopadajícího světla se na povrchu sledovaného předmětu obvykle mění pomalu a plynule. Oproti tomu odrazivost povrchu se často mění rychle a skokově. Pokud je navíc sledovaný povrch předmětu dostatečně hladký, je možné lineárním filtrováním snímku $I'(\mathbf{x})$ oddělit tyto dvě složky [15], a tím se zbavit vlivů osvětlení.

1.3 Detekce změny

Detekce změny zajišťuje vytvoření masky $M^{(2)}$ na základě detekovaných změn mezi snímky I_1 a I_2 . Při detekci změny je třeba se vypořádat se všemi vlivy, které nebyly

odstraněny předchozími bloky. Patří mezi ně šum kamery, chyby registrace, perspektivní deformace, zbylé rozdíly v intenzitách, chyby zaostření kamery, rozostření pohybem, chyby způsobené omezeným rozlišením kamery a jiné.

Výběr metody pro detekci změny není přímočarý. V každém případě je nutné uvažovanou metodu otestovat na reálných datech a výsledky porovnat s jinými metodami.

1.3.1 Rozdílová metoda

Rozdílová metoda [15] (differencing) je jedna z nejjednodušších, ale i nejpoužívanějších metod detekce změny. Definujme funkce $I_1(\mathbf{x})$ a $I_2(\mathbf{x})$ udávající intenzity snímků I_1 resp. I_2 a funkci $M^{(2)}(\mathbf{x})$ udávající hodnoty masky $M^{(2)}$. Pro detekování rozdílů mezi těmito snímky nejprve stanovme rozdílový snímek

$$D(\mathbf{x}) = I_1(\mathbf{x}) - I_2(\mathbf{x}). \quad (1.31)$$

Výsledná hodnota masky $M^{(2)}$ je pak dána:

$$M^{(2)}(\mathbf{x}) = \begin{cases} 1 & \text{pokud } |D(\mathbf{x})| > \tau_d, \\ 0 & \text{jinak.} \end{cases} \quad (1.32)$$

Hodnota prahu τ_d je obvykle určena empiricky.

1.3.2 Poměr intenzit a metoda stínovacího modelu

Metoda poměřující intenzity pixelů snímku ve zlomku je založená na velice podobném principu jako rozdílová metoda. Nejdříve ze snímků $I_1(\mathbf{x})$ a $I_2(\mathbf{x})$ stanovme poměrový snímek

$$P(\mathbf{x}) = \frac{I_1(\mathbf{x})}{I_2(\mathbf{x})}. \quad (1.33)$$

Výsledná maska $M^{(2)}$ je pak dána:

$$M^{(2)}(\mathbf{x}) = \begin{cases} 1 & \text{pokud } |P(\mathbf{x}) - 1| > \tau_p, \\ 0 & \text{jinak.} \end{cases} \quad (1.34)$$

Hodnota prahu τ_p je opět volena empiricky.

Metoda stínovacího modelu. Metoda stínovacího modelu [20] (shading model method) určuje masku $M^{(2)}$ opět z poměrového snímku $P(\mathbf{x})$. Využívá však při tom zjednodušeného modelu stínování dle vzorce 1.30. Tento vztah je opět rozdělen na část i_L , závisící převážně na osvětlení, a část i_S určenou materiálem sledovaného předmětu:

$$L_d(\lambda) = i_L(\lambda)i_S(\lambda), \quad (1.35)$$

kde $i_L(\lambda) = L_i(\lambda)[\mathbf{v}_i \cdot \mathbf{n}]^+$ a $i_S(\lambda) = f_d(\lambda)$. Pokud mezi snímky nedošlo ke strukturální změně, poměr intenzit pixelů ze snímků, pro určitou vlnovou délku λ , se redukuje na:

$$P(\mathbf{x}) = \frac{I_1(\mathbf{x})}{I_2(\mathbf{x})} = \frac{L_{d1}(\mathbf{x})}{L_{d2}(\mathbf{x})} = \frac{i_{L1}(\mathbf{x})}{i_{L2}(\mathbf{x})}. \quad (1.36)$$

Předpokládejme, že hodnota funkce $i_{Lj}(\mathbf{x})$ je hladká a pomalu se mění v závislosti na vektoru \mathbf{x} ($j \in \{1, 2\}$). Rozptyl hodnot funkce $P(\mathbf{x})$ na malé spojitě oblasti Ω_i okolo \mathbf{x}_i bude nyní přibližně nulový. Pokud ovšem došlo ke strukturální změně, rozptyl bude značně větší. Výsledná maska $M^{(2)}$ je dána jako:

$$M^{(2)}(\mathbf{x}_i) = \begin{cases} 1 & \text{pokud } \sum_{\mathbf{x} \in \Omega_i} (P(\mathbf{x}) - \mu_i)^2 > \tau_s, \\ 0 & \text{jinak.} \end{cases} \quad (1.37)$$

μ_i je průměrná hodnota $P(\mathbf{x})$ na oblasti Ω_i . Hodnota prahu τ_s a velikost oblasti Ω_i je volena empiricky.

1.3.3 Metoda LCP

Metody založené na LCP [3] určují změnu pixelu na poloze \mathbf{x}_i na základě čtvercového okolí $\rho(\mathbf{x}_i)$ tohoto pixelu ve snímku I_1 a čtvercového okolí $\pi(\mathbf{x}_i + \Delta\mathbf{x})$ ve snímku I_2 . S použitím korelační funkce $Corr(\rho, \pi)$ pak pro každý pixel \mathbf{x}_i stanovíme hodnotu podobnosti jako:

$$S(\mathbf{x}_i) = \max_{\Delta\mathbf{x} \in \mathcal{D}} \{Corr(\rho(\mathbf{x}_i), \pi(\mathbf{x}_i + \Delta\mathbf{x}))\}, \quad (1.38)$$

kde $\mathcal{D} = \{(x, y) \in \mathbb{Z}^2 : -\delta_c \leq x, y \leq \delta_c\}$. Výhodou tohoto přístupu je, že dokáže potlačit chyby registrace.

Jako korelační funkci lze použít například *normovanou křížovou korelaci* [3] (normed cross correlation):

$$Corr_{cc}(\rho, \pi) = \frac{\sum_{\mathbf{x}_i \in \rho, \mathbf{y}_i \in \pi} I_1(\mathbf{x}_i) I_2(\mathbf{y}_i)}{\sqrt{\sum_{\mathbf{x} \in \rho} I_1(\mathbf{x})^2 \sum_{\mathbf{x} \in \pi} I_2(\mathbf{x})^2}}, \quad (1.39)$$

nebo *normovaný součet kvadratických odchylek*³ [3] (normed sum of squared differences):

$$Corr_{sd}(\rho, \pi) = \frac{\sum_{\mathbf{x}_i \in \rho, \mathbf{y}_i \in \pi} (I_1(\mathbf{x}_i) - I_2(\mathbf{y}_i))^2}{\sqrt{\sum_{\mathbf{x} \in \rho} I_1(\mathbf{x})^2 \sum_{\mathbf{x} \in \pi} I_2(\mathbf{x})^2}}. \quad (1.40)$$

Výsledná maska se poté opět vytvoří prahováním:

$$M^{(2)}(\mathbf{x}_i) = \begin{cases} 1 & \text{pokud } S(\mathbf{x}_i) < \tau_c, \\ 0 & \text{jinak.} \end{cases} \quad (1.41)$$

³Pokud použijeme funkci $Corr_{sd}$ je nutné ve vzorci 1.38 vyhledávat místo maxima minimum a ve vzorci 1.41 obrátit znaménko $<$ na $>$.

Hodnota prahu τ_c a hodnota δ_c je volena empiricky. Čím větší je hodnota δ_c , s tím větší chybou registrace se tato metoda může vypořádat. Nastavení příliš velké hodnoty však může zakrýt některé malé změny, které ve snímku nastaly. Velikost oblastí ρ a π se obvykle volí srovnatelná s rozměry změněných oblastí.

1.3.4 Metoda LBP/LTP histogramů

Pro účely této práce jsme navrhli novou metodu detekující změny na základě LBP (Local Binary Patterns) nebo LTP (Local Ternary Patterns) deskriptorů. Před samotnou detekcí změn pro každý bod na snímku I_1 a I_2 vypočteme LBP či LTP deskriptory, čímž získáme dvourozměrné pole D_1 a D_2 . Tato pole následně srovnáme bod po bodu porovnáním histogramů vytvořených z jejich okolí. Tedy např. pro detekci změny v bodě \mathbf{x} nejprve určíme histogramy $H_1(D_1, \mathcal{R})$ a $H_2(D_2, \mathcal{R})$, $\mathcal{R} = \{\mathbf{x} + (\delta_x \ \delta_y)^T : (\delta_x \ \delta_y)^T \in \mathbb{Z}^2 \wedge -\delta_h \leq \delta_x, \delta_y \leq \delta_h\}$. Porovnáním těchto histogramů pak dostaneme hodnotu $S(\mathbf{x})$ vyjadřující podobnost bodu \mathbf{x} mezi snímky I_1 a I_2 . Prahováním této hodnoty použitím prahu τ_h získáme výslednou masku:

$$M^{(2)}(\mathbf{x}_i) = \begin{cases} 1 & \text{pokud } S(\mathbf{x}_i) > \tau_h, \\ 0 & \text{jinak.} \end{cases} \quad (1.42)$$

Velikost oblasti \mathcal{R} je dána hodnotou δ_h a měla by přibližně odpovídat velikosti změněné oblasti v pixelech. Tento princip se často označuje jako metoda *klouzajícího okna* (sliding window).

Díky využití LBP či LTP deskriptorů je navržená metoda nezávislá na osvětlení sledovaného předmětu. Při porovnávání pomocí histogramů jsou do určité míry kompenzovány chyby registrace a chyby způsobené paralaxou. Ze své podstaty tato metoda vychází z textury snímku. Pokud tedy nastane změna která se neprojevuje změnou textury, nebude tato metoda pracovat správně. Vzhledem k charakteru vstupních snímků však v této práci zmíněné omezení nečiní problémy.

LBP

Metoda LBP [8] umožňuje přidělit každému pixelu vstupního snímku deskriptor popisující texturu v jeho okolí. Seskupením deskriptorů z určité oblasti snímku je pak možné například identifikovat předměty na snímku na základě textury. Tato metoda je velmi často používána pro rozpoznávání obličejů.

Metoda je založená na prahování sousedních pixelů hodnotou centrálního pixelu. Výsledná hodnota deskriptoru je pak dána zřetazením binárního výsledku tohoto prahování do binárního řetězce. Originální metoda pracuje s osmi sousedními pixely, a výsledkem je tedy osmibitový deskriptor. Označíme-li si intenzitu centrálního pixelu jako $I(\mathbf{x})$ a intenzity jednotlivých okolních pixelů jako $I(g(\mathbf{x}, i))$, $i \in$

$\{0, 1, \dots, 7\}$, výsledný deskriptor pixelu je dán sumou:

$$D(\mathbf{x}) = \sum_{i=0}^7 s(I(g(\mathbf{x}, i)) - I(\mathbf{x})) \cdot 2^i, \quad (1.43)$$

kde

$$s(x) = \begin{cases} 1 & \text{pokud } x \geq 0 \\ 0 & \text{pokud } x < 0. \end{cases} \quad (1.44)$$

Jednotlivé elementy textury jako jsou hrany, rohy, body a podobně jsou ve výsledném deskriptoru reprezentovány určitou posloupností bitů. Tyto posloupnosti se nazývají *vzory* (patterns).

Uniformní vzory. Pro osmibitové deskriptory existuje celkem 256 různých vzorů. Histogram vytvořený z těchto deskriptorů by se skládal z 256 sloupců. Kvůli zvýšení efektivity srovnávání těchto histogramů je často využíváno tzv. *uniformních vzorů* [12]. Uniformní vzory jsou vzory, ve kterých se vyskytuje dvojice po sobě jdoucích bitů 01 či 10 maximálně dvakrát. Například v řetězci 00011000 se vyskytuje jedna dvojice 01 a jedna dvojice 10, řetězec je tedy uniformní. Řetězec 00100001 však již uniformní není, jelikož se zde vyskytuje dvakrát dvojice bitů 01 a dvakrát dvojice 10 (povšimněte si, že počítáme také dvojici mezi prvním a posledním bitem). Pro osmibitový deskriptor existuje celkem 58 uniformních vzorů. Všechny ostatní vzory jsou ve výsledku mapovány do jednoho sloupce histogramu, dostaneme tedy histogram s 59 sloupci. Kromě snížení počtu sloupců histogramu vykazují uniformní vzory také menší citlivost na šum.

Zobecnění LBP. Oproti původnímu návrhu je ve zobecněné verzi LBP (generic LBP) [12] deskriptor vypočten porovnáním intenzity centrálního bodu $I(\mathbf{x})$ s intenzitami bodů rozmístěných na kružnici okolo \mathbf{x} (viz obrázek 1.5). Poloměr této kružnice r_l je volitelný stejně jako počet bodů N_l na ní rozmístěných. Poznamenejme, že $g'(\mathbf{x}, i) = (g_1 \ g_2)^T$ jsou nyní reálnými vektory ($g'(\mathbf{x}, i) \in \mathbb{R}^2$):

$$g'(\mathbf{x}, i) = \mathbf{x} + [r \cdot \cos(i \cdot 2\pi/(N-1)) \quad r \cdot \sin(i \cdot 2\pi/(N-1))]^T \quad (1.45)$$

Intenzitu okolních bodů $I'(g'(\mathbf{x}, i)), i \in \{0, 1, \dots, N\}$ je tedy nutné určit pomocí bilineární interpolace.

$$I'(g'(\mathbf{x}, i)) = b_{11}I(\mathbf{q}_{11}) + b_{12}I(\mathbf{q}_{12}) + b_{21}I(\mathbf{q}_{21}) + b_{22}I(\mathbf{q}_{22}), \quad (1.46)$$

kde $\mathbf{q}_{kl} = (x_k \ y_l)^T$ a $x_1 = \lfloor g_1 \rfloor$, $x_2 = \lceil g_1 \rceil$, $y_1 = \lfloor g_2 \rfloor$, $y_2 = \lceil g_2 \rceil$. Koeficienty b_{mn} pak určíme jako

$$\begin{aligned} b_{11} &= \frac{(x_2 - g_1)(y_2 - g_2)}{(x_2 - x_1)(y_2 - y_1)} & b_{12} &= \frac{(x_2 - g_1)(g_2 - y_1)}{(x_2 - x_1)(y_2 - y_1)} \\ b_{21} &= \frac{(g_1 - x_1)(y_2 - g_2)}{(x_2 - x_1)(y_2 - y_1)} & b_{22} &= \frac{(g_1 - x_1)(g_2 - y_1)}{(x_2 - x_1)(y_2 - y_1)} \end{aligned} \quad (1.47)$$

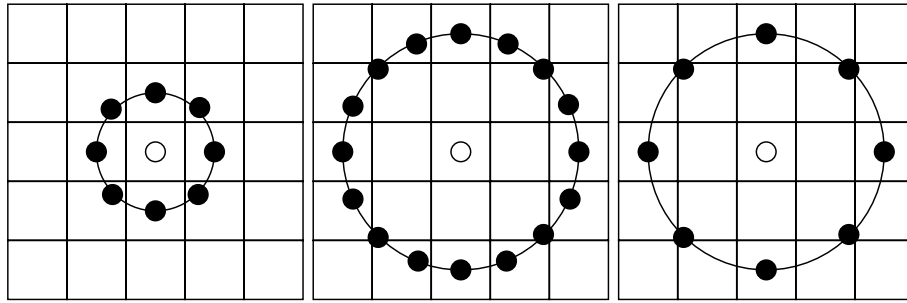
Výsledný deskriptor pak vypočteme dle vzorce:

$$D(\mathbf{x}) = \sum_{i=0}^{N-1} s(I'(g'(\mathbf{x}, i)) - I(\mathbf{x})) \cdot 2^i, \quad (1.48)$$

kde

$$s(x) = \begin{cases} 1 & \text{pokud } x \geq 0 \\ 0 & \text{pokud } x < 0. \end{cases} \quad (1.49)$$

Zvolíme-li například $N_l = 8$ bude výsledný deskriptor osmibitový stejně jako u originální metody. Výhodou tohoto rozšíření je menší citlivost na šum, jelikož jsou porovnávány intenzity určené interpolací ze čtyř pixelů.



Obr. 1.5: Zobecnění LBP deskriptoru. Na obrázku vidíte několik možností konfigurace okolních bodů u zobecněné verze LBP. Nastavení parametrů je pro levý obrázek $N_l = 8$, $r_l = 1$, pro prostřední $N_l = 16$, $r_l = 2$ a pro pravý obrázek $N_l = 8$, $r_l = 2$.

LTP

LTP [8] je modifikací LBP vykazující menší citlivost na šum. Použitím prahu τ_l jsou okolní pixely dle hodnoty $x = I(g(\mathbf{x}, i)) - I(\mathbf{x})$, $i = \{0, 1, \dots, 7\}$ zařazeny do tří různých kategorií:

$$s'(x) = \begin{cases} -1 & \text{pokud } x < -\tau_l \\ 1 & \text{pokud } x > \tau_l \\ 0 & \text{jinak.} \end{cases} \quad (1.50)$$

Takto získáme posloupnost:

$$p = \{s'(I(g(\mathbf{x}, i)) - I(\mathbf{x}))\}_{i=0}^7 \quad (1.51)$$

Vytvoření binárního deskriptoru ze třístavových elementů této posloupnosti již není tak intuitivní. V této práci jsme LTP posloupnost p rozdělili na dvě LBP posloupnosti (kladnou a zápornou) jejichž zřetěžením jsme pak získali 16bitový deskriptor. Uvažujme nyní že jsme pomocí LTP určili následující posloupnost:

$$p = \{-1, -1, 0, 0, -1, 1, 1, 1\}. \quad (1.52)$$

Tuto posloupnost nyní rozdělíme na dvě části:

$$\begin{aligned} p^+ &= \{p[i]^+\}_{i=0}^7 \\ p^- &= \{p[i]^-\}_{i=0}^7, \end{aligned} \quad (1.53)$$

kde

$$p[i]^+ = \begin{cases} 1 & \text{pokud } p[i] > 0 \\ 0 & \text{jinak} \end{cases} \quad p[i]^- = \begin{cases} 1 & \text{pokud } p[i] < 0 \\ 0 & \text{jinak.} \end{cases} \quad (1.54)$$

$p[i]$ zde reprezentuje i -tý element posloupnosti p . Získáme tak posloupnosti

$$\begin{aligned} p^+ &= \{0, 0, 0, 0, 0, 1, 1, 1\}, \\ p^- &= \{1, 1, 0, 0, 1, 0, 0, 0\}. \end{aligned} \quad (1.55)$$

Zřetezením těchto posloupností a následným převedením na binární řetězec pak již získáme výsledný 16bitový deskriptor: 0000011111001000.

Uniformní vzory. Koncepti uniformních vzorů je možné rozšířit také na LTP. Pokud se v posloupnosti p daného vzoru vyskytuje po sobě jdoucí dvojice elementů $\{0, 1\}$, $\{1, 0\}$, $\{0, -1\}$, $\{-1, 0\}$, $\{1, -1\}$ a $\{-1, 1\}$ nanejvýš dvakrát, označíme vzor jako uniformní. Například posloupnost

$$\{0, 0, -1, -1, -1, 0, 0, 0\}, \quad (1.56)$$

je uniformní, protože se v něm vyskytuje jedna dvojice $\{0, -1\}$ a jedna dvojice $\{-1, 0\}$. Posloupnost

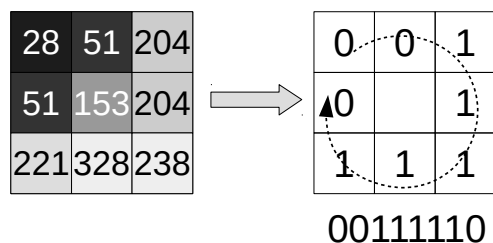
$$\{-1, -1, 0, 0, 0, 1, 1, 1\} \quad (1.57)$$

však uniformní není, jelikož se zde vyskytuje dvojice $\{-1, 0\}$, $\{0, 1\}$ a $\{1, -1\}$ (opět počítáme také dvojici tvořenou prvním a posledním elementem). Mapováním neuniformních vzorů do jednoho sloupce histogramu pak získáme celkem 172 sloupců histogramu.

Robustní LTP. Zavedením prahu τ_l jsme sice získali deskriptor s menší citlivostí na šum, ovšem tímto přístupem je také limitována nezávislost na osvětlení scény. Ve článku [25] je tento problém řešen použitím tzv. robustních LTP deskriptorů. Jedná se o modifikaci LTP, kde je práh τ_l pro každý pixel snímku vypočten jako $\tau_l = \alpha \cdot \mu_x$. Střední hodnota μ_x je určena jako:

$$\mu_x = \frac{1}{9}(I(\mathbf{x}) + \sum_{i=0}^7 I(g(\mathbf{x}, i))) \quad (1.58)$$

a α je volitelná konstanta.



Obr. 1.6: Tvorba LBP deskriptoru. Vlevo je zobrazen centrální pixel a jeho osm okolních pixelů (hodnoty intenzity jsou zapsány dekadicky). Prahováním okolních pixelů získáme binární hodnoty (vpravo). Průchodem po směru hodinových ručiček pak zřetězíme tyto binární hodnoty a získáme osmibitový deskriptor: 00111110.

Zobecnění LTP. Metodu LTP je možné opět zobecnit na stejném principu jako LBP. Výsledkem bude deskriptor s menší citlivostí na šum s možností volby velikosti okolí a počtu porovnávaných bodů.

Porovnání histogramů

Pro porovnání histogramů jsme v této práci použili χ^2 vzdálenost [16]. Uvažujme nyní dva histogramy H_1 a H_2 . Hodnota $H_i(j)$, udává pravděpodobnost výskytu vzoru odpovídajícího sloupci j histogramu i . Pro určení podobnosti těchto dvou histogramů pak slouží tento vzorec:

$$S_{\chi^2}(H_1, H_2) = \sum_{j=1}^N \frac{[H_1(j) - H_2(j)]^2}{[H_1(j) + H_2(j)]}, \quad (1.59)$$

kde N je počet sloupců histogramu.

2 NÁVRH A IMPLEMENTACE ŘEŠENÍ

Pro implementaci vybraných metod jsme využili programovací jazyk C++ a knihovnu OpenCV¹. Implementace metody pro detekci změny navíc využívá knihovnu TinyXML² pro načítání konfiguračních souborů testovacích databází. K sestavení zdrojových kódů jsme použili multiplatformního systému CMAKE³. Zdrojové kódy jsou dostupné v příloze A.3. Bližší informace o návrhu a provedení metod jsou uvedeny v následujících odstavcích.

2.1 Detekce změny

2.1.1 Volba metod pro detekci změny

V tomto kroku se pokusíme vizuálním srovnáním výsledků popsanych metod rozhodnout které metody případně kombinace metod má smysl dále testovat. K těmto účelům jsme zvolili dvojici snímků pořízených statickou kamerou. Tyto snímky již byly představeny v úvodu (viz obrázek 3).

Rozdílová metoda. Maska získaná rozdílovou metodou je zobrazena na obrázku 2.1. Je zde vidět že kromě správně označené oblasti změny metoda navíc vyznačuje velké množství pixelů, kde ke změně nedošlo. Můžeme předpokládat, že tento problém je způsobený chybou registrace.

Poměr intenzit. Maska získaná touto metodou je zobrazena na obrázku 2.2. Na výsledné masce lze opět vidět chyby z důvodu nepřesné registrace, ale přidává se k nim navíc chyba způsobená osvětlením. Karta která byla následně ze skříně vyjmuta zřejmě vytvářela stín. Při vyjmutí tento stín zmizel a došlo ke změně osvětlení zadní stěny skříně. Přítomnost této chyby naprosto odpovídá vzorci 1.36, ze kterého vychází metoda stínovacího modelu.

Metoda stínovacího modelu. Maska získaná touto metodou je na obrázku 2.3. Zdá se, že pro naše účely tato metoda není vhodná. Předpoklad „hladké a pomalu se měnící“ funkce $i_{Lj}(\mathbf{x})$ (odstavec 1.3.2, Metoda stínovacího modelu) zřejmě není splněn. Tato funkce totiž závisí také na normálových vektorech osvětleného povrchu. Na hladkých plochách metoda funguje správně ovšem na nerovnostech chybně detekuje změny.

¹<http://opencv.org/>

²<http://www.grinninglizard.com/tinyxml/>

³<https://cmake.org/>

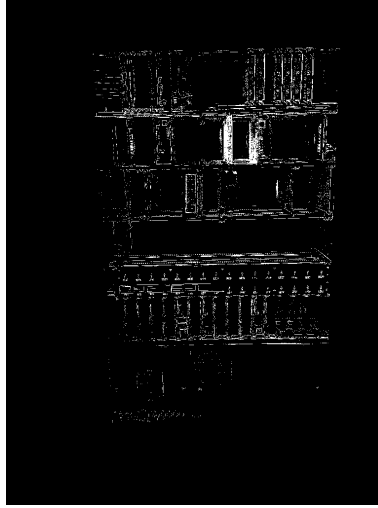
Metoda LCP. Maska získaná touto metodou je na obrázku 2.4. Metoda za použití obou korelačních funkcí vykazuje dobré výsledky. Zdá se, že výborně potlačuje chyby registrace. Detekovaná oblast změny je však oproti skutečnosti větší.

Kombinace rozdílové metody a metody LCP. Pohledem na výsledky rozdílové metody a metody LCP lze usoudit, že kombinací těchto dvou metod bychom mohli dosáhnout dobrých výsledků. Obě metody správně označily změněnou oblast, avšak mimo tuto oblast vyznačily množství dalších pixelů. Provedeme-li kombinaci masek získaných diferenční metodou a metodou LCP operací AND, získáme masku na obrázku 2.5 (ponecháme tedy v masce $M^{(2)}$ vyznačené pouze oblasti, které označily za změněné obě tyto metody). Takto navrženou metodu budeme nyní značit jako LCP-D.

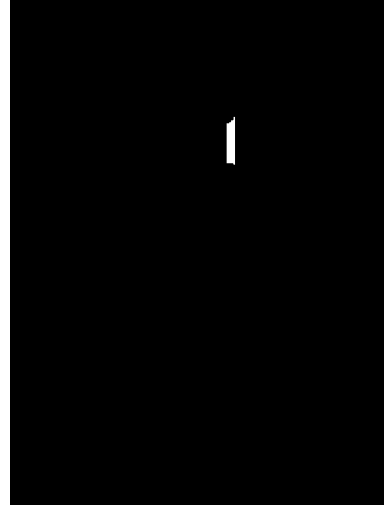
Metoda LBP histogramů. V této práci jsme množství vzorů redukovali na uniformní vzory tak, jak je popsáno v sekci 1.3.4. Masku získanou touto metodou je na obrázku 2.6. Chyby registrace jsou zjevně velmi dobře potlačeny. Ovšem vidíme zde hodně chyb způsobených šumem. Na hodnotách podobnostního snímku $S(\mathbf{x})$ lze také pozorovat, že poměr požadovaného signálu k šumu je velmi malý.

Metoda LTP histogramů. V této práci jsme použili modifikaci LTP, a to zobecněné robustní LTP. Tyto modifikace jsou popsány v sekci 1.3.4. Počet vzorů byl opět redukován na uniformní vzory. Masku získanou touto metodou je na obrázku 2.7. Poměr signálu k šumu je oproti LBP mnohem větší. Chyby registrace jsou opět velmi dobře potlačeny.

Výsledek srovnání. Ze všech uvedených metod se na první pohled nejlépe osvědčila metoda LCP-D a metoda LTP histogramů. Proto budeme v dalším textu probírat implementaci a přesnost právě těchto dvou metod.

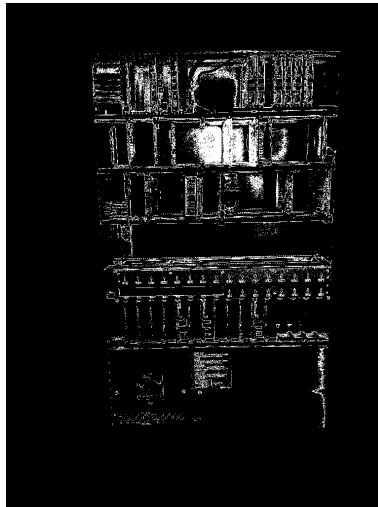


(a) Maska $M^{(2)}$

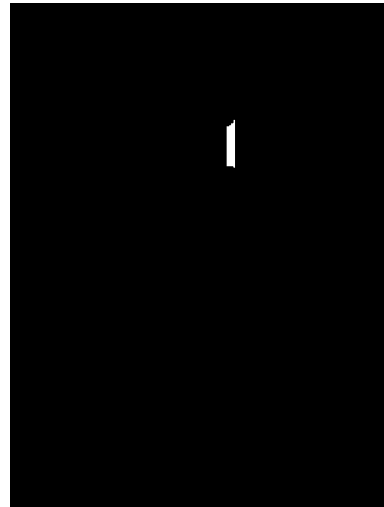


(b) Maska $G^{(2)}$

Obr. 2.1: Výstup rozdílové metody. Obrázek (a) zobrazuje změny detekované rozdílovou metodou. Obrázek (b) zobrazuje masku $G^{(2)}$. Pro získání této masky byl práh τ_d nastaven na hodnotu 0,2.

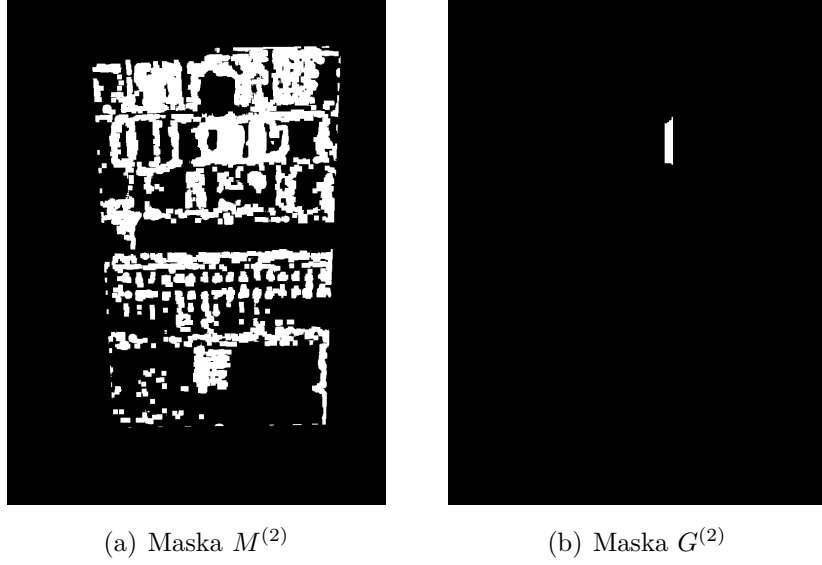


(a) Maska $M^{(2)}$

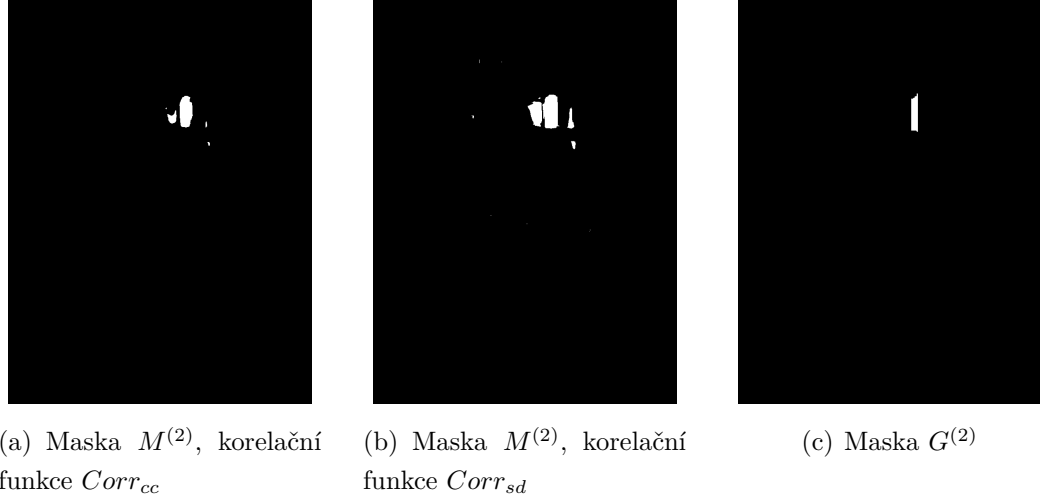


(b) Maska $G^{(2)}$

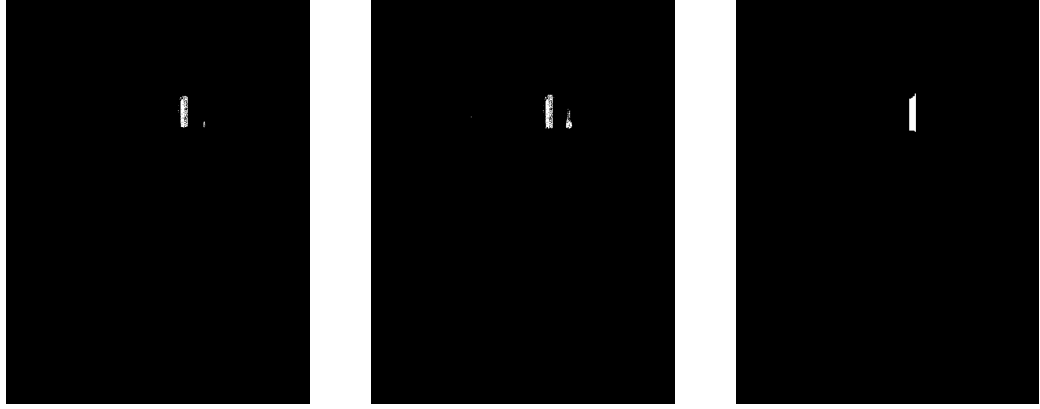
Obr. 2.2: Výstup metody poměru intenzit. Obrázek (a) zobrazuje změny detekované metodou poměru intenzit. Obrázek (b) zobrazuje masku $G^{(2)}$. Pro získání této masky byl práh τ_p nastaven na hodnotu 0,5.



Obr. 2.3: Výstup metody stínovacího modelu. Obrázek (a) zobrazuje změny detekované metodou stínovacího modelu. Obrázek (b) zobrazuje masku $G^{(2)}$. Pro získání této masky byl práh τ_s nastaven na hodnotu 0,5 a velikost oblasti Ω_i byla 10×10 pixelů.



Obr. 2.4: Výstup metody LCP. Obrázek (a) zobrazuje změny detekované metodou LCP za použití $Corr_{cc}$ jako měřítka podobnosti. Obrázek (b) zobrazuje změny detekované metodou LCP za použití $Corr_{sd}$. Obrázek (c) zobrazuje masku $G^{(2)}$. Pro získání této masky byl práh τ_c nastaven na hodnotu 0,5 a hodnota δ_c na hodnotu 6 pixelů.

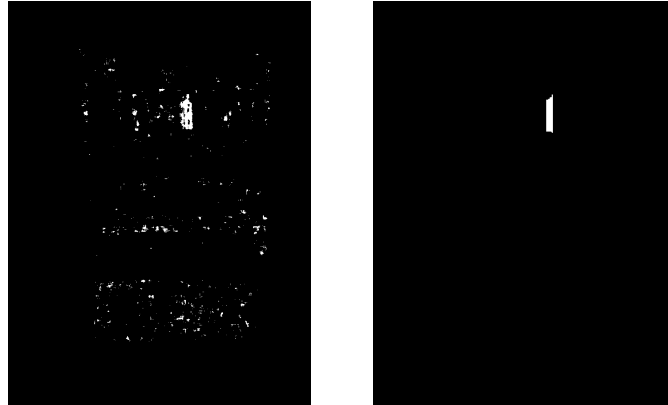


(a) Maska $M^{(2)}$, korelační
funkce $Corr_{cc}$

(b) Maska $M^{(2)}$, korelační
funkce $Corr_{sd}$

(c) Maska $G^{(2)}$

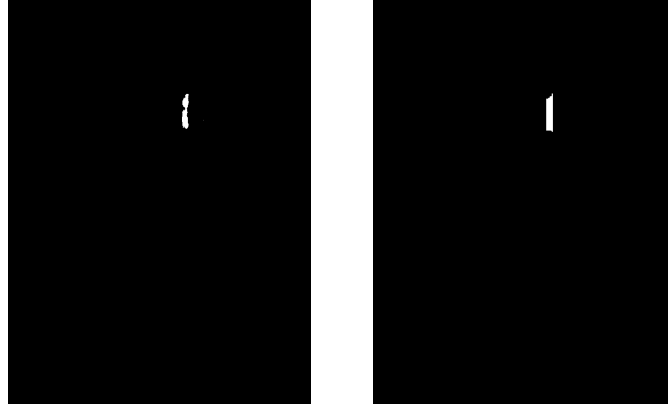
Obr. 2.5: Výstup metody LCP-D. Obrázek (a) zobrazuje změny detekované za použití $Corr_{cc}$ jako měřítka podobnosti. Obrázek (b) zobrazuje změny detekované za použití $Corr_{sd}$. Obrázek (c) zobrazuje masku $G^{(2)}$. Pro získání těchto masek byly parametry nastaveny takto: $\tau_d = 0,32$, $\tau_c = 0,5$, $\delta_c = 6$.



(a) Maska $M^{(2)}$

(b) Maska $G^{(2)}$

Obr. 2.6: Výstup metody LBP. Obrázek (a) zobrazuje změny detekované za použití LBP. Obrázek (c) zobrazuje masku $G^{(2)}$. Pro získání těchto masek byly parametry nastaveny takto: $\tau_h = 70$, $\delta_h = 5$.



(a) Maska $M^{(2)}$

(b) Maska $G^{(2)}$

Obr. 2.7: Výstup metody LTP. Obrázek (a) zobrazuje změny detekované za použití LTP (zobecněné robustní LTP). Obrázek (c) zobrazuje masku $G^{(2)}$. Pro získání těchto masek byly parametry nastaveny takto: $r_l = 2$, $N_l = 8$, $\alpha = 0,07$, $\tau_h = 130$ a $\delta_h = 5$.

2.1.2 Metoda LCP-D

Výsledná implementace této metody se skládá ze čtyř kroků:

- **Předzpracování.** V tomto kroku jsou vstupní obrázky I_1 a I_2 převedeny na odstíny šedi.
- **Registrace.** Pro registraci jsme nejprve použili metodu klíčových bodů ORB. Získané klíčové body jsme pak spárovali metodou popsanou v sekci 1.1.1. Pro určení odhadu matice H_{12} jsme pak použili funkci **findHomography** z OpenCV. Tento odhad je následně upřesněn metodou ECC.
- **Normalizace.** Pro normalizaci jsme použili globální metodu popsanou v sekci 1.2.1.
- **Detekce.** V tomto kroku jsme výslednou masku $M^{(2)}$ určili jako

$$M^{(2)}(\mathbf{x}) = M_D^{(2)}(\mathbf{x}) \cdot M_C^{(2)}(\mathbf{x}), \quad (2.1)$$

kde $M_D^{(2)}$ je binární maska získaná rozdílovou metodou a $M_C^{(2)}$ je maska získaná metodou LCP.

V kroku detekce změny jsou testovány dvě různé korelační funkce a to $Corr_{cc}$ a $Corr_{sd}$ (viz sekce 1.3.3).

2.1.3 Metoda LTP histogramů

Výsledná implementace této metody se skládá ze tří kroků:

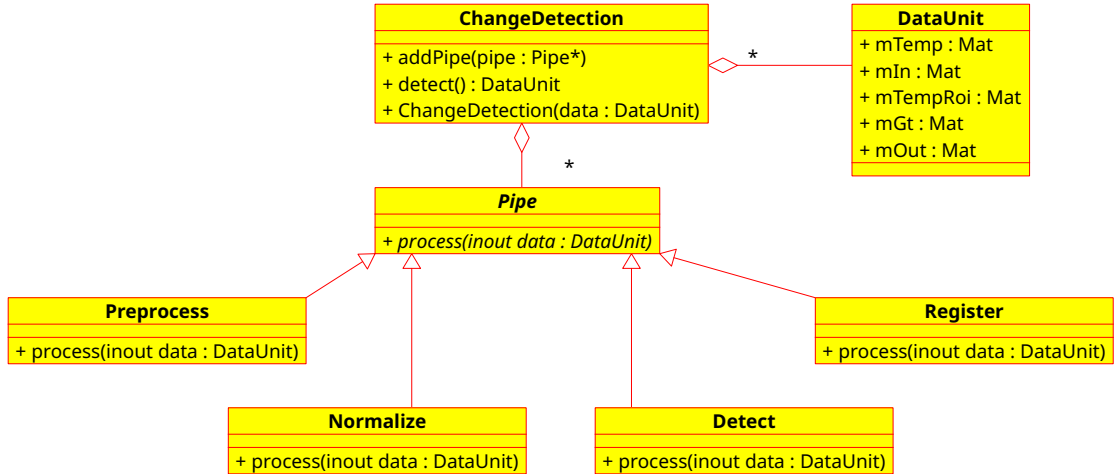
- **Předzpracování.** V tomto kroku jsou nejprve vstupní obrázky I_1 a I_2 převedeny na odstíny šedi a dále jsou filtrovány gaussovým filtrem pro potlačení šumu.
- **Registrace.** Pro registraci jsme nejprve použili metodu klíčových bodů ORB. Získané klíčové body jsme pak spárovali metodou popsanou v sekci 1.1.1. Pro určení odhadu matice H_{12} jsme pak použili funkci **findHomography** z OpenCV. Tento odhad je následně upřesněn metodou ECC.
- **Detekce.** V tomto kroku jsme použili implementaci zobecněných robustních LTP příznaků (výslednou metodu lze tedy parametrizovat hodnotami r_l , N_l , α , τ_h a δ_h). Pro výpočet histogramů v klouzajícím okně jsme z důvodu efektivity výpočtu použili implementaci *integrálních histogramů* [13]. Výsledné histogramy jsou pak porovnány pomocí χ^2 vzdálenosti (viz sekce 1.3.4).

Krok normalizace v tomto případě není nutný, detekce změny je provedena pouze na základě textury. Při testování jsme dokonce zjistili, že pokud je normalizace provedena jsou výsledky detekce dokonce horší.

2.1.4 Objektový návrh

Konečný návrh je zde popsán z hlediska objektového programování, viz UML(Unified Modeling Language) diagram na obrázku 2.8. Tento návrh je společný pro obě metody detekce změny, liší se pouze výsledná implementace uvedených tříd. Třída **ChangeDetection** je kontejnerem objektů typu **Pipe**. Tato třída provádí detekci změny na základě jednotlivých kroků, které implementují třídy **Preprocess**, **Register**, **Normalize** a **Detect**. Abstraktní třída **Pipe** pouze popisuje rozhraní jednoho kroku. Vstupní i výstupní snímky, na kterých tyto třídy pracují, jsou uloženy ve struktuře **DataUnit**. Tuto strukturu si udržuje instance **ChangeDetection** a při provádění detekce (metoda **detect**) jí postupně předává všem objektům **Pipe**, které ji mohou změnit. Tento návrh byl zvolen z důvodu jednoduché implementace a snadného rozšiřování. Provedení jednotlivých tříd je popsáno dále. Zdrojový kód programu je dostupný v příloze A.3.

Třída ChangeDetection. Tato třída provádí celý proces detekce změny. Konstruktor **ChangeDetection** vytváří instanci třídy a zároveň uloží do členské proměnné objekt **DataUnit**, který obsahuje veškeré snímky, se kterými bude instance pracovat. Metoda **addPipe** přidává do členské fronty objekty typu **Pipe**. Metoda **detect** postupně volá metodu **process** každého objektu ve frontě (tedy provádí jednotlivé kroky detekce) a předává jí snímky ve formě **DataUnit**. Volaná metoda tyto snímky zpracuje a upraví. Po dokončení posledního kroku detekce, metoda **detect** vrátí volajícímu zpracované snímky **DataUnit**.



Obr. 2.8: UML diagram konečného návrhu.

Struktura DataUnit. Tato struktura obsahuje veškeré vstupní i výstupní snímky, které jsou při detekci změny potřeba. Jejimi členskými proměnnými jsou tyto snímky:

- **mTemp** – referenční snímek I_1 ,
- **mIn** – zpracovávaný snímek I_2 ,
- **mTempRoi** – binární maska R ,
- **mGt** – binární maska $G^{(2)}$,
- **mOut** – výstupní binární maska $M^{(2)}$.

Třída Preprocess. Tato třída provádí předzpracování snímků. Přesná implementace se u obou detekčních metod liší a je provedena, tak jak bylo uvedeno v sekcích 2.1.2 a 2.1.3.

Třída Register. Tato třída provádí registraci snímků I_1 a I_2 . Registrace je pro obě navržené detekční metody provedena použitím metody klíčových bodů v kombinaci s metodou ECC. Přesná implementace je popsána v sekcích 2.1.2 a 2.1.3.

Třída Normalize. Tato třída provádí normalizaci intenzit snímků. Pro metodu LTP histogramů není normalizace použita. Přesná implementace této třídy je tedy popsána v sekci 2.1.2.

Třída Detect. Tato třída provádí detekci změny mezi snímky a vytváří tak masku $M^{(2)}$. Ve výsledném programu je tato třída implementována hned několikrát, a to pro každou z testovaných metod. Všechny tyto metody jsou teoreticky popsány v sekci

1.3. Konečná implementace je pak popsána v sekci 2.1.2 a 2.1.3. Názvy těchto tříd sdílí společnou předponu **Detect-**. Jde o třídy:

- **DetectDiff** – implementuje detekci změny rozdílovou metodou,
- **DetectCorr** – implementuje detekci změny metodou LCP, s oběma popsánými korelačními funkcemi,
- **DetectDiffCorr** – implementuje detekci změny metodou LCP-D,
- **DetectLTP** – implementuje detekci změny pomocí LTP. histogramů.

2.2 Nalezení úhlu pohledu

Výpočet homografie. Navržené řešení pro hledání úhlu pohledu je teoreticky popsáno v sekci 1.1.3. Pro určení vektoru \mathbf{v}' je třeba nejprve vypočítat matici homografie $\tilde{\mathbf{H}}_N$. V naší implementaci jsme k tomuto účelu využili metodu klíčových bodů ORB. Stejně jako u úlohy č.1 jsme opět spárovali klíčové body metodou, jež je popsána v sekci 1.1.1. Pomocí funkce **findHomography** z OpenCV vypočteme pro každý snímek F_N odhad homografie $\widehat{\mathbf{H}}_N$ představující transformaci $I_1 \rightarrow F_N$. Pro lepší stabilitu výsledné homografie jsme použili sekvenční metodu, kde výsledná matice homografie vychází ze všech odhadů $\{\widehat{\mathbf{H}}_1, \widehat{\mathbf{H}}_2, \dots, \widehat{\mathbf{H}}_N\}$, dle rekursivního vzorce:

$$\tilde{\mathbf{H}}_N = \frac{1}{2}(\tilde{\mathbf{H}}_{N-1} + \widehat{\mathbf{H}}_N) \quad (2.2)$$

Před použitím tohoto vzorce jsou však nejprve vyřazeny odhady $\widehat{\mathbf{H}}_N$, které se příliš liší od matice $\tilde{\mathbf{H}}_{N-1}$. Pravidlo pro vyřazování odhadů jsme definovali na základě *spektrální normy* [1], jako:

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}\mathbf{A}^H)}, \quad (2.3)$$

kde $\lambda_{\max}(\mathbf{A}\mathbf{A}^H)$ označuje největší vlastní číslo matice $\mathbf{A}\mathbf{A}^H$ a \mathbf{A}^H označuje Hermiteovsky sdruženou matici k \mathbf{A} . Pokud platí

$$\|\widehat{\mathbf{H}}_N - \tilde{\mathbf{H}}_{N-1}\|_2 < \tau_n, \quad (2.4)$$

je výsledná matice $\tilde{\mathbf{H}}_N$ určena dle vzorce 2.2. V opačném případě zůstává matice stejná jako v předchozím kroku $\tilde{\mathbf{H}}_N = \tilde{\mathbf{H}}_{N-1}$. Hodnotu práhu τ_n je třeba zvolit s ohledem na maximální rychlost pohybu kamery.

Signalizace. Z důvodu signalizace vykresluje na každém novém snímku F_N kříž složený ze dvou přímkou procházejících středem snímku. Na pozici dané vektorem \mathbf{n} pak vykresluje kružnici signalizující směr a velikost odchylky úhlu pohledu. Tento vektor je dán vzorcem:

$$\mathbf{n} = s\mathbf{v}' + [0, 5 \cdot \mathbf{m}], \quad (2.5)$$

kde s je volitelná konstanta ovlivňující citlivost signalizace a \mathbf{m} je vektor udávající rozměry snímku F_N v pixelech.



Obr. 2.9: Ukázka signalizace na símku F_N . Červená kružnice uprostřed snímku signalizuje odchylku úhlu pohledu.

3 VÝSLEDKY PRÁCE

3.1 Vyhodnocení přesnosti zvolených metod

Pro posouzení přesnosti navržených metod detekce změny společně s metodou pro hledání úhlu pohledu jsme použili pět testovacích databází. Na těchto databázích jsme pak provedli detekci změny pomocí navržených metod a jejich výstup porovnali s ručně vytvořenými maskami $G^{(2)}$.

3.1.1 Testovací databáze

První dvě databáze byly pořízeny 8megapixelovou kamerou v tabletu *Samsung Galaxy Tab S*.

- Databáze *rack_01* obsahuje 6 snímků, porovnávat však budeme jen páry, mezi kterými nedošlo k pohybu kamery. Tím získáme 4 páry snímků.
- Databáze *rack_02* obsahuje 5 snímků. Zde budeme porovnávat každý snímek s každým, s výjimkou snímků, které se nevyznačují žádnou strukturální změnou. Získáme tím 8 párů snímků. Mezi snímky v této databázi dochází k pohybu kamery.

Následující tři databáze byly pořízeny méně kvalitní 1megapixelovou webkamerou *Creative Live!*.

- Databáze *rack_03* obsahuje 6 snímků, pořízených staticky postavenou kamerou. Porovnávat budeme referenční snímek s ostatními. Získáme tedy celkem 5 párů snímků.
- Databáze *rack_04* obsahuje 6 snímků, pořízených pohyblivou kamerou. Porovnávat budeme referenční snímek s ostatními. Získáme tak celkem 5 párů snímků.
- Databáze *rack_05* obsahuje 6 snímků pořízených pohyblivou kamerou. Při pořizování této databáze jsme využili navrženou metodu pro vyhledávání úhlu pohledu. Porovnávat budeme referenční snímek s ostatními. Získáme tak celkem 5 párů snímků.

Všechny zmíněné databáze jsou uvedeny v příloze A.1.

3.1.2 Měření přesnosti

Pro měření přesnosti metody budeme porovnávat jednotlivé pixely masky $M^{(2)}$ a masky $G^{(2)}$. Předpokládejme, že pixely s intenzitou 1 v těchto maskách označují změnu a ostatní pixely mají intenzitu 0. Základními měřítky pro nás budou tyto

hodnoty:

$$\begin{aligned}
TP &= \sum_{\mathbf{x} \in \mathcal{M}} M^{(2)}(\mathbf{x})G^{(2)}(\mathbf{x}), \\
FP &= \sum_{\mathbf{x} \in \mathcal{M}} M^{(2)}(\mathbf{x})|G^{(2)}(\mathbf{x}) - 1|, \\
TN &= \sum_{\mathbf{x} \in \mathcal{M}} |M^{(2)}(\mathbf{x}) - 1||G^{(2)}(\mathbf{x}) - 1|, \\
FN &= \sum_{\mathbf{x} \in \mathcal{M}} |M^{(2)}(\mathbf{x}) - 1|G^{(2)}(\mathbf{x}),
\end{aligned} \tag{3.1}$$

kde \mathcal{M} je množina všech pixelů daných masek. Hodnota TP (True Positive) zde odpovídá počtu pixelů, které byly správně označeny jako změněné. Hodnota FP (False Positive) odpovídá počtu pixelů, které byly chybně označeny jako změněné. Hodnota TN (True Negative) odpovídá počtu pixelů, které byly správně označeny jako nezměněné. Konečně hodnota FN (False Negative) odpovídá počtu pixelů, které byly chybně označeny jako nezměněné. Pro vyhodnocení přesnosti metody na dané databázi jsou všechny tyto hodnoty sečteny přes všechny páry snímků databáze.

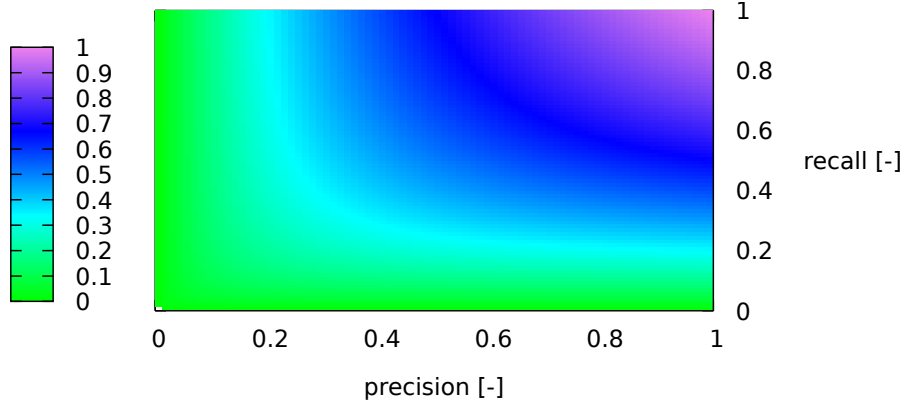
Z těchto hodnot pak určíme tzv. *f-measure* [3], podle kterého budeme hodnotit přesnost navržených metod.

$$FM = \frac{2 \cdot Pr \cdot Re}{Pr + Re}, \tag{3.2}$$

kde $Pr = \frac{TP}{TP+FP}$ a $Re = \frac{TP}{TP+FN}$.

Hodnota Pr zvaná *precision* udává jaké množství¹ pixelů, klasifikovaných jako změněné, bylo skutečně změněno. Hodnota Re zvaná *recall* udává jaké množství¹ skutečně změněných pixelů bylo metodou identifikováno. V ideálním případě, kdy metoda pracuje naprosto dokonale, bude Pr a Re nabývat hodnoty 1. V reálné situaci však tyto hodnoty budou menší. Nastavení parametrů metody je optimální, jsou-li hodnoty Pr a Re přibližně stejné. Hodnota FM je s touto podmínkou optimality konzistentní, nabývá totiž nejvyšší hodnoty, jsou-li obě hodnoty (Pr a Re) stejné, viz obrázek 3.1.

¹Toto množství neudává počet pixelů, jde o relativní hodnotu.



Obr. 3.1: Závislost hodnoty f-measure na hodnotách precision a recall.

3.1.3 Vyhodnocení

Z výsledného grafu (viz obrázek 3.2) plyne, že nejvyšší přesnost detekce změny jsme dosáhli metodou LTP histogramů, a to na databázi pořízené statickou kamerou ($FM = 0,812$). Metoda LTP histogramů dosahuje na většině databázích větší přesnost než LCP-D. U některých snímků se však projevuje nedostatek textury chybou detekce změny. Tento problém je však možné řešit vyplněním uzavřených oblastí ve výsledné masce $M^{(2)}$, jelikož metoda LTP histogramů bez problémů detekuje hraniční body změněné oblasti.

Metoda LCP-D je zjevně velmi citlivá na kvalitu použité kamery. Na databázích pořízených méně kvalitní kamerou klesá její přesnost na $FM = 0,1$. Na databázích pořízených kvalitnější kamerou je přesnost metody pro obě korelační funkce přibližně stejná.

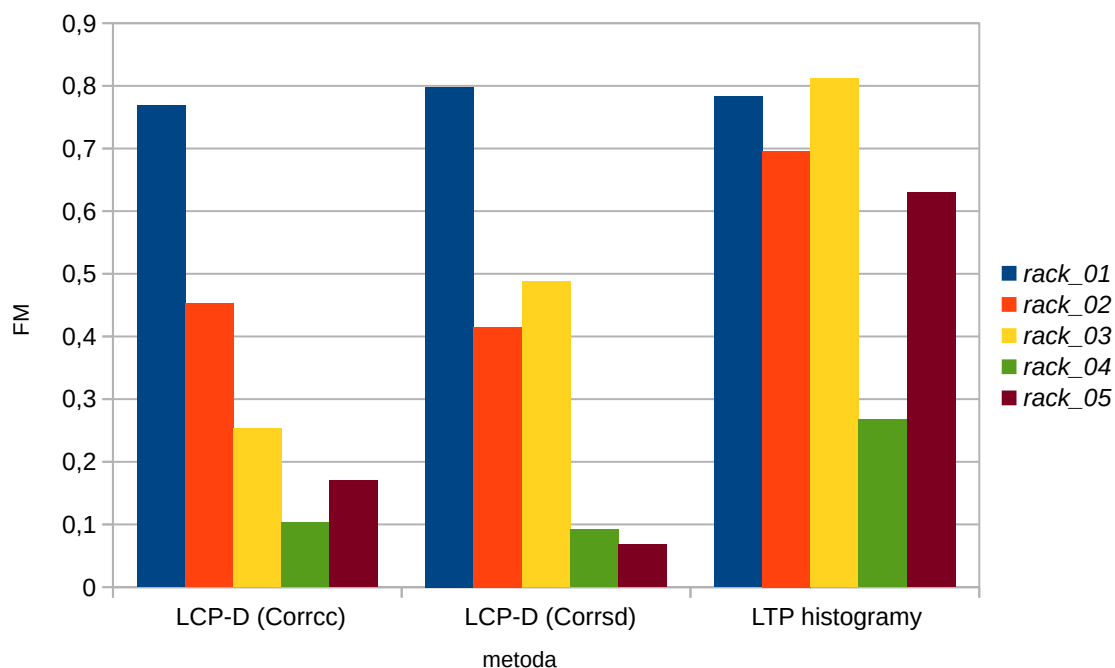
Vyhodnocení přesnosti zvolených metod na testovacích databázích je zapsáno v tabulce 3.2. Při testování metod jsme použili dvě množiny parametrů (podle použité kamery). Tyto parametry jsou uvedeny v tabulce 3.1.

databáze	ρ, π	δ_c	τ_s	τ_d	r_l	N_l	α	τ_h	δ_h
<i>rack_01, rack_02</i>	16×16	6	0,48	0,32	2	8	0,07	130	5
<i>rack_03, rack_04, rack_05</i>	16×16	6	0,35	0,25	2	8	0,07	400	10

Tab. 3.1: Parametry metod použité při testování.

metoda	LCP-D ($Corr_{cc}$)			LCP-D ($Corr_{sd}$)			LTP histogramy		
databáze	Pr	Re	FM	Pr	Re	FM	Pr	Re	FM
<i>rack_01</i>	0,497	0,417	0,453	0,335	0,546	0,415	0,777	0,629	0,696
<i>rack_02</i>	0,950	0,646	0,769	0,895	0,721	0,798	0,812	0,757	0,783
<i>rack_03</i>	0,855	0,148	0,253	0,572	0,426	0,488	0,934	0,718	0,812
<i>rack_04</i>	0,076	0,161	0,103	0,052	0,410	0,092	0,163	0,759	0,268
<i>rack_05</i>	0,235	0,134	0,170	0,038	0,329	0,068	0,544	0,748	0,630

Tab. 3.2: Vyhodnocení zvolených metod na testovacích databázích.



Obr. 3.2: Porovnání přesnosti testovaných metod na testovacích databázích.

4 ZÁVĚR

V této práci jsme prozkoumali některé techniky detekce změn v digitálních obrazech včetně problematik které detekci provázejí (registrace a normalizace intenzit). Navrhli jsme novou metodu pro detekci změny založenou na LTP histogramech. Navrženou metodu jsme porovnali s šesti dalšími metodami pro detekci změny. Následně jsme metodu LTP histogramů společně s jinou vybranou metodou (LCP-D) podrobili rozsáhlejšímu testování na pěti databázích reálných snímků.

Na většině testovacích databázích dosahuje metoda LTP histogramů lepších výsledků než metoda LCP-D. Nejvyšší dosaženou hodnotu f-measure ($FM = 0,812$) jsme získali metodou LTP histogramů, a to na snímcích pořízených statickou kamerou.

Kromě metody pro detekci změny jsme v této práci navrhli metodu pro vyhledávání úhlu pohledu (z důvodu minimalizace chyby způsobené paralaxou). Rozdíl mezi přesností metody LTP histogramů na databázích *rack_04* a *rack_05* potvrzuje účinnost této metody. Na databázi *rack_05*, pořízené za použití metody pro vyhledávání úhlu, je přesnost větší.

Při řešení této práce jsme také implementovali metodu integrálních histogramů [13], která dosud nebyla dostupná v OpenCV. Tuto implementaci jsme zveřejnili¹ pod licencí Apache 2.0².

¹<https://github.com/deu439/Integral-Histograms>

²<http://www.apache.org/licenses/LICENSE-2.0.html>

LITERATURA

- [1] ARRÓSPIDE, J., et al. Homography-based ground plane detection using a single on-board camera. In: *IET Intelligent Transport Systems*. 2010, 4(2), s. 149-. DOI: 10.1049/iet-its.2009.0073. ISSN 1751956x.
- [2] AACH, Til – KAUP, André – RUDOLF, Mester. Statistical model-based change detection in moving video. Elsevier. *Signal Processing*, 1993, vol. 31, no. 2, s. 165–180.
- [3] BENEDEK, Csaba. *Novel markovian change detection models in computer vision*. Budapest, 2008. PhD thesis on the Pázamány Péter Catholic University.
- [4] CALONDER, Michael, et al. *BRIEF: Binary Robust Independent Elementary Features*. Lausanne, Switzerland: EPFL, CvLab.
- [5] EVANGELIDIS, Georgios D. – PSARAKIS, Emmanouil Z. Parametric image alignment using enhanced correlation coefficient maximization. IEEE Computer Society. *IEEE transactions on pattern analysis and machine intelligence*. October 2008, vol. 30, no. 10.
- [6] HARTLEY, Richard – ZISSERMAN, Andrew. *Multiple view geometry in computer vision*. 2nd ed. New York: Cambridge University Press, 2003. ISBN 0-521-54051-8.
- [7] HUA, Gang – BROWN, Matthew – WINDER, Simon. Discriminant embedding for local image descriptors. In: *International Conference on Computer Vision*. Rio de Janeiro, Brazil: October 14-20, 2007.
- [8] HUANG, Di, et al. Local Binary Patterns and Its Application to Facial Image Analysis: A Survey. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2011, 41(6), s. 765-781. DOI: 10.1109/TSMCC.2011.2118750. ISSN 1094-6977.
- [9] LOWE, David G. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision*. Hingham, MA, USA: Kluwer Academic Publishers, 2004.
- [10] MIKOLAJCZYK, Krystian – SCHMID, Cordelia. A Performance Evaluation of Local Descriptors. IEEE Computer Society. *IEEE transactions on pattern analysis and machine intelligence*. October 2005, vol. 27, no. 10. p. 1615–1630.

- [11] NIELSEN, Allan – CANTY, Morton. Kernel principal component analysis for change detection. In *Image and Signal Processing for Remote Sensing. XIV.* Cardiff, Wales, United Kingdom: SPIE, 2008.
- [12] PIETIKÄINEN, Matti. *Computer vision using local binary patterns.* London: Springer, 2011. Computational imaging and vision, v. 40. ISBN 978-0-85729-747-1.
- [13] PORIKLI, Fatih. Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces. In *EEE International Conference on Computer Vision and Pattern Recognition (CVPR).* San Diego, CA, USA: IEEE, 2005.
- [14] PRIMDAHL, Keith – KATZ, et al. Change detection from multiple camera images extended to non-stationary cameras. In *Proceedings of Field and Service Robotics.* 2005.
- [15] RADKE, Richard J., et al. *Image change detection algorithms: a systematic survey.* Rensselaer Polytechnic Institute. 110 8th Street, Troy, NY, 12180 USA: Rensselaer Polytechnic Institute, 2004.
- [16] RASSEM, Taha H. – KHOO, Bee Ee. *Completed Local Ternary Pattern for Rotation Invariant Texture Classification.* Engineering Campus, Nibong Tebal, 14300 Penang, Malaysia: School of Electrical & Electronic Engineering, Universiti Sains Malaysia.
- [17] ROSTEN, Edward – DRUMMOND, Tom. *Machine learning for high-speed corner detection.* United Kingdom, Cambridge: University of Cambridge.
- [18] RUBLEE, Ethan, et al. ORB: an efficient alternative to SIFT or SURF. In *International Conference on Computer Vision.* Fira de Barcelona, Barcelona, Spain: Alberto Sanfeliu, 2011. p. 2564-2571.
- [19] SHAKHNAROVICH, Gregory. *Learning task-specific similarity.* Massachusetts, USA, 2005. PhD thesis on the Massachusetts Institute of Technology.
- [20] SKIFSTAD, Kurt – JAIN, Ramesh. Illumination independent change detection for real world image sequences. In *Computer vision, graphics, and image processing.* Academic Press Professional, Inc. San Diego, CA, USA: 1989 p. 387–399.
- [21] ST-CHARLES, Pierre-Luc – BILODEAU, Guillaume-Alexandre. Improving background subtraction using Local Binary Similarity Patterns. In *Applications of Computer Vision.* Steamboat Springs, CO, USA: IEEE 2014.

- [22] SZELISKI, Richard. *Computer Vision: Algorithms and Applications*. September 3, 2010 draft. United States, New York City: Springer, 2010.
- [23] WIEMKER, Rafael, et al. Unsupervised robust change detection on multispectral imagery using spectral and spatial features. In *Third International Airborne Remote Sensing Conference and Exhibition*. Environmental Research Institute of Michigan. Copenhagen, Denmark: Environmental Research Institute of Michigan, 1997.
- [24] WANG, Yi – LUO, Zhiming – JODOIN, Pierre-Marc. Interactive deep learning method for segmenting moving objects. In *Pattern Recognition Letters*. 2016.
- [25] WU, Xiaosheng, et al. Improved Local Ternary Patterns for Automatic Target Recognition in Infrared Imagery. In *Sensors*. St. Alban-Anlage 66, 4052 Basel, Switzerland: MDPI AG, 2015.

SEZNAM ZKRATEK

BRDF	Bidirectional Reflectance Distribution Function
BRIEF	Binary Robust Independent Elementary Features
ECC	Enhanced Correlation Coefficient maximization
FAST	Features from Accelerated Segment Test
HOG	Histogram of Oriented Gradients
PCA	Principal Component Analysis
ID3	Iterative Dichotomiser 3
LBP	Local Binary Patterns
LCP	Local Correlation Peak
LCP-D	kombinace metody LCP a rozdílové metody
LDA	Linear Discriminant Analysis
LTP	Local Ternary Patterns
MB-LBP	Macro-Block LBP
MB-LTP	Macro-Block LTP
ORB	Oriented BRIEF
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
UML	Unified Modeling Language

SEZNAM SYMBOLŮ

$Corr_{cc}$	korelační funkce na bázi normované křížové korelace
$Corr_{sd}$	korelační funkce na bázi normovaného součtu kvadratických odchylek
F_N	aktuální snímek z kamery
$G^{(2)}$	ručně vytvořená binární maska označující oblasti změny
$G^{(n)}$	ručně vytvořená n-ární maska označující umístění a typ změny
$\tilde{\mathbf{H}}_{12}$	matice homografie mezi snímky I_1 a I_2
$\tilde{\mathbf{H}}_N$	matice homografie mezi snímky I_1 a F_N
I_1	referenční snímek
I_2	zpracovávaný snímek
r_l	poloměr zobecněné verze LBP/LTP
D_1	dvourozměrné pole příznaků vypočtených na referenčním snímku
D_2	dvourozměrné pole příznaků vypočtených na zpracovávaném snímku
$M^{(2)}$	výstupní binární maska vyznačující detekované oblasti změny
$M^{(n)}$	výstupní n-ární maska vyznačující detekované umístění a typ změny
N_l	počet okolních bodů zobecněné verze LBP/LTP
R	binární maska oblasti zájmu
α	koeficient určující práh robustní verze LTP
δ_c	rozkmít metody LCP
δ_h	rozkmít metody LBP/LTP histogramů
π	oblast metody LCP
ρ	oblast metody LCP
τ_c	práh metody LCP
τ_d	práh rozdílové metody
τ_h	práh metody LCP/LTP histogramů

τ_p	práh metody poměru intenzit
τ_s	práh metody stínovacího modelu
Ω_i	oblast metody stínovacího modelu

SEZNAM PŘÍLOH

A	Obsah elektronické přílohy	57
A.1	Složka „databáze“	57
A.2	Složka „vyhodnocení“	57
A.3	Složka „zdrojové kódy“	57

A OBSAH ELEKTRONICKÉ PŘÍLOHY

K této práci jsme přiložili elektronickou přílohu ve formě zip souboru. Tato příloha obsahuje databázi testovacích snímků ve složce „databáze“, výsledné masky $M^{(2)}$ získané testovanými metodami ve složce „vyhodnocení“ a zdrojové kódy navrženého programu ve složce „zdrojové kódy“. Z důvodu omezení velikosti přílohy informačním systémem VUT jsme byli nuceni elektronicky odevzdanou verzi přílohy zredukovat. Tato verze přílohy tedy obsahuje pouze zdrojové kódy programu.

A.1 Složka „databáze“

V podsložkách „rack_01“, „rack_02“, „rack_03“, „rack_04“ a „rack_05“ jsou všechny testovací databáze. Snímky s příponou „_GT“ jsou masky $G^{(2)}$, snímky s příponou „_ROI“ jsou masky R , ostatní snímky jsou použity jako I_1 nebo jako I_2 . Pro detekci změny jsou jednotlivé snímky asociovány pomocí konfiguračního souboru „data.xml“.

A.2 Složka „vyhodnocení“

V podsložkách „ccorr“ a „sqdiff“ jsou výsledky detekce změny na testovacích databázích s použitím metody LCP-D („ccorr“ odpovídá použití funkce $Corr_{cc}$ a „sqdiff“ použití $Corr_{sd}$). V podsložce „ltp“ jsou pak výsledky detekce změny na testovacích databázích s použitím metody LTP histogramů.

A.3 Složka „zdrojové kódy“

Tato složka obsahuje kompletní zdrojové kódy programu. V podsložce „detekce změny“ jsou zdrojové kódy metody pro detekci změny. V podsložce „nalezení úhlu“ jsou zdrojové kódy metody pro vyhledávání úhlu pohledu. Pro kompilaci programu je třeba mít nainstalovaný systém CMAKE, knihovnu OpenCV a knihovnu TinyXML.